Grouped EbE方式用前処理付き反復法の計算量削減

柴田, 修作

藤野, 清次
九州大学情報基盤研究開発センター

https://doi.org/10.15017/20055

出版情報：九州大学大学院システム情報科学紀要. 16 (2), pp.63-68, 2011-09-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第 16 巻　第 2 号　平成 23 年 9 月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.16, No.2, September 2011

Grouped EbE 方式用前処理付き反復法の計算量削減

柴田　修作* ・ 藤野　清次**

Reduction of Computational Cost of Iterative Method Preconditioned for Grouped EbE Scheme

Shusaku SHIBATA* and Seiji FUJINO**

(Received July 22, 2011)

Abstract: Large scale analysis with Finite Element Method (FEM) needs much memory because of construction of a global matrix. This motivates us to employ Element by Element (EbE) scheme, which does not require to construct global matrix. However, in the EbE scheme, general useful preconditioners for global matrix cannot be applied. To enhance effectiveness of the EbE scheme, variants of the EbE schemes, such as Grouped EbE, Clustered EbE, and preconditioners for them have been proposed by many researchers. In the Grouped EbE scheme, parallel implementation of the EBE scheme becomes easier. The clustered EbE scheme, which is regarded as a generalization of standard EbE scheme, can decrease iterations. Therefore, in this paper, we propose a technique for the purpose of reduction of computational cost of preconditioning used in the Grouped EbE scheme, and demonstrate its effectiveness.

Keywords: Preconditioning, EbE scheme, CEbE scheme, GEbE scheme, Eisenstat trick

1. は じ め に

有限要素法 (Finite Element Method)(以下，FEM と略す) による解析では，Element-by-Element 方式 (以下，EbE 方式と略す) によって，全体行列を構成せずに解析を行う EbE 型 FEM が用いられてきた[1]．さらに，EbE-Gauss Seidel 型前処理 (以下，EbE-GS 型前処理と略す) のような EbE 方式の特徴に合わせた EbE 方式用の前処理も用いられた[2–4]．近年の並列計算技術の発展に伴い，EbE 型 FEM にも高並列性が求められるようになった[5]．しかし，EbE 方式用の前処理は互いに依存関係を持つ要素間で逐次処理を必要とするため並列性が低い．そこで，高並列な Grouped EbE 方式 (以下，GEbE 方式と略す) が提案された[6, 7]．同時に，GEbE-GS 型前処理などの GEbE 方式用の前処理も提案された．さらに，EbE 方式の一般化形として Clustered EbE 方式 (以下，CEbE 方式と略す) が提案された[8]．

本研究では，前処理付き反復法の計算量削減手法である Eisenstat 技法[9, 10]の考え方を応用し，GEbE-GS 型前処理付き反復法の計算量削減手法として GEbE-Eisenstat 技法を提案し，その有効性を明らかにする．

本論文は次のように構成される．第 2 節では，EbE, GEbE, CEbE 方式について記述する．第 3 節では，GS 型前処理[11]と GEbE-GS 型前処理について記述する．第 4 節では，全体行列用の計算量削減手法 Eisenstat 技法について記述する．第 5 節では，GEbE 方式用の計算量削減手法 GEbE-Eisenstat 技法を提案する．第 6 節では，数値実験を通して提案手法の効果と有用性を明らかにする．第 7 節でまとめを行う．

平成 23 年 7 月 22 日受付
*情報学専攻修士課程
**情報基盤研究開発センター

2. FEM における要素記憶方式

FEM では，解析領域を多数の部分領域に分割し，各領域内において単純な関数を用いて未知量を近似する．この部分領域のことを要素と呼び，分割によって生じた格子点を節点と呼ぶ．通常，節点には節点番号が与えられ，各要素の頂点も，要素内において要素節点番号を与えられる．複数の異なる要素の頂点が，同一の節点番号を持つ場合もある．

FEM では，有限要素方程式と呼ばれる小さな離散化方程式を要素毎に作成し，それらを重ね合わせて解析領域全体の離散化方程式を得る．例えば，各要素の頂点の自由度が 1 である場合，ある解析領域を ne 個の要素で分割し，N 個の節点が生じたとき，最終的に得られる離散化方程式の近似解は，以下の連立一次方程式の求解で得られる．

$$A\boldsymbol{x} = \boldsymbol{b}. \tag{1}$$

ここで，係数行列 A を次元数 $N \times N$ の実数行列，$\boldsymbol{x}$ を N 次元の解ベクトル，$\boldsymbol{b}$ を N 次の右辺ベクトルとする．係数行列 A は全体行列と呼ばれ，

$$A = \sum_{i=1}^{ne} A_i^e \tag{2}$$

と表せる．要素行列 A_i^e は i 番の要素行列を示す．要素行列とは，有限要素方程式において現れる係数行列である．要素行列は全体行列と同様に $N \times N$ の行列である．要素行列の各列と行は，要素の頂点に各々対応している．例えば，頂点の自由度が 1 である要素 e において，この要素の頂点 1，2，3 が節点番号 j，$j+1$，$j+4$ $(1 \leq j \leq N-4)$ を与えられている場合，要素行列 A^e は j，$j+1$，$j+4$ 番目の行と列上にのみ非零成分を持つ．

互いに同じ節点番号を与えられた頂点を持つ要素行列が複数存在する場合，全体行列 A を作成するための重ね合わせの際に，全体行列上の同一の位置に，異なる複数の要素行列の非零成分が重ね合わせられることがある．この様な関係にある複数の要素行列は，共有する節点において互いに依存関係にある．

本節では，全体行列を構成せずに解析を行うための三種類の方式について記述する．EbE 方式，GEbE 方式，CEbE 方式と各々呼ばれる．これらの方式は要素の記憶方法に以下のような特徴を持つ．

(1) **EbE**：単一の要素を別個に憶える[1)]
(2) **GEbE**：節点共有のない要素をまとめて憶える[7)]
(3) **CEbE**：隣接する要素をまとめて憶える[8)]

本論文では，各方式を**要素記憶方式**と総称する．

2.1 EbE 方式

EbE 方式では，全体行列 A は構成されず，要素行列 A^e のみを憶える．また，N_i^e を要素 e_i における (頂点数) × (頂点の自由度) の値とすると，要素行列 A_i^e は $N_i^e \times N_i^e$ の密行列として憶える．

2.2 GEbE 方式

GEbE 方式では，互いに節点を共有しない要素同士を集め，要素集合を作る．$\mathcal{G}$ は全要素を含む集合，$\mathcal{G}_k$ は k 番の要素集合，ng は要素集合数 (number of groups) を示す．

(1) $\mathcal{G} = \bigcup_{k=1}^{ng} \mathcal{G}_k, \quad \phi = \mathcal{G}_j \bigcap \mathcal{G}_k \quad \text{for } j \neq k.$
(2) 各要素集合内の要素間では，節点は共有されない．
(3) 要素集合を形成する際，ng の値を最小にする．

要素集合 $\mathcal{G}_k$ 内の要素行列を重ね合わせ，集合行列 (group matrix)A_k が形成される．このとき，集合行列と全体行列 A は次のように定義される．

$$A_k = \sum_{e \in \mathcal{G}_k} A^e \quad (k = 1, 2, \ldots, ng), \quad A = \sum_{k=1}^{ng} A_k. \tag{3}$$

2.3 CEbE 方式

CEbE 方式では，互いに隣接し辺を共有する要素を集め，群を形成する．この群を**隣接要素群**と呼ぶ．$\mathcal{C}$ は全ての要素を含む集合，$\mathcal{C}_j$ は j 番の隣接要素群内の要素の集合を各々示す．nc は隣接要素群の数 (number of cluster) を示す．

$$\mathcal{C} = \bigcup_{j=1}^{nc} \mathcal{C}_j, \quad \phi = \bigcap_{j=1}^{nc} \mathcal{C}_j.$$

このとき，隣接要素群 $\mathcal{C}_j$ 内の各要素の要素行列 A^e を重ね合わせ，隣接要素群行列 (cluster matrices)A_j が形成される．また，隣接要素群行列と全体行列 A は

$$A_j = \sum_{e \in \mathcal{C}_j} A^e, \quad A = \sum_{j=1}^{nc} A_j. \tag{4}$$

と定義される．

Fig. 1 は，三角形要素 e_2 を中心とする隣接要素群の一例を示す．各隣接要素群内の要素数 (number of elements in a cluster) を nec で表す．nec の値を 1 から 4 まで変化させたときの各隣接パターンを描いた．

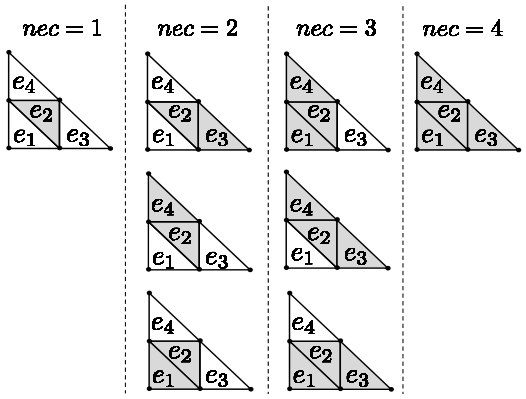

Fig. 1　Four examples of a cluster involving element e_2.

2.4 CEbE 方式と他の要素記憶方式の関係

CEbE 方式では，隣接要素群を一つの要素と見做すと，GEbE 方式の考え方を適用することができる．言い換えると，CEbE 方式と GEbE 方式は併用可能である．各隣接要素群内の要素数：nec = 1 のとき，CEbE 方式は EbE 方式と等しい．したがって，CEbE 方式は EbE 方式を一般化したものである[8)]．すなわち，CEbE 方式は EbE 方式を内包する，と考えることが出来る．

3. 要素記憶方式別の前処理

一般的な前処理は全体行列に対して適用されるが，EbE，GEbE，CEbE 方式に対して適用することは出来ない．そこで，各要素記憶方式に対応する形で作られた前処理を用いる必要がある．例えば，同じ GS 型前処理でも，全体行列の要素記憶方式に応じて前処理行列の作り方が異なる．本論文ではこの区分を，全体行列用の前処理，**EbE 方式用の前処理**，**GEbE 方式用の前処理**，**CEbE 方式用の前処理**と

表す．本節では，全体行列を構成する場合と要素記憶方式を用いる場合で分けた，以下 2 種類の前処理について記述する．その後，要素記憶方式と前処理についてその概要を記述する．

(1) GS：全体行列用の前処理[11)]
(2) **GEbE-GS：GEbE 方式用の前処理**[6, 7)]

3.1 GS 型前処理

L，D，U を式 (1) で示した全体行列 A の狭義下三角行列，対角行列，狭義上三角行列とする．A を

$$A = L + D + U \tag{5}$$

と分離する．このとき，対角スケーリングを施した $\tilde{A}$ の前処理行列 $\tilde{K}$ は以下のように表せる．

$$\tilde{K} = (\tilde{L} + I)(\tilde{U} + I). \tag{6}$$

3.2 GEbE-GS 型前処理

L_k，D_k，U_k を式 (3) で示した集合行列 A_k の狭義下三角行列，対角行列，狭義上三角行列とする．A_k を

$$A_k = L_k + D_k + U_k \tag{7}$$

と分離する．このとき，対角スケーリングを施した $\tilde{A}$ の前処理行列 $\tilde{M}_{GEbE}$ は以下のように表せる．

$$\tilde{M}_{\mathrm{GEbE}} = \prod_{k=1}^{ng} (\tilde{L}_k + I) \prod_{k=ng}^{1} (\tilde{U}_k + I). \tag{8}$$

4. 全体行列用の計算量削減手法

本節では，全体行列用の計算量削減手法として，**Eisenstat 技法**[9, 10)]について述べる．以下，GS 型前処理付き反復法に対する Eisenstat 技法の適用方法と効果を記述する．式 (6) に基づいて，全体行列の両側から GS 型前処理を適用した反復法における全体行列とベクトルの積計算は

$$\boldsymbol{g} = (\tilde{L} + I)^{-1} \tilde{A} (\tilde{U} + I)^{-1} \boldsymbol{v} \tag{9}$$

と表す．$\tilde{A}$ は対角スケーリングを適用した全体行列を各々示す．このとき，式 (5) より，$\tilde{A}$ を

$$\tilde{A} = \tilde{L} + \tilde{D} + \tilde{U} = \tilde{L} + I + \tilde{U} \tag{10}$$

と分離する．ただし，$\tilde{D}$ は $\tilde{A}$ の対角行列を示す．式 (10) を式 (9) に代入して，以下に示す．

$$\boldsymbol{g} = (\tilde{L} + I)^{-1} \{\tilde{L} + I + \tilde{U}\} (\tilde{U} + I)^{-1} \boldsymbol{v}. \tag{11}$$

Eisenstat 技法では，式 (11) に示す行列ベクトル積の計算量を削減する．

4.1 行列ベクトル積の計算量

以下では，GS 型前処理における式 (11) の計算量について記述する．ベクトル$\boldsymbol{p}$，$\boldsymbol{q}$を

$$\boldsymbol{p} = (\tilde{U} + I)^{-1} \boldsymbol{v}, \tag{12}$$

$$\boldsymbol{q} = (\tilde{L} + I + \tilde{U}) \boldsymbol{p} \tag{13}$$

と定義し，ベクトル$\boldsymbol{q}$を用いて，式 (11) を

$$\boldsymbol{g} = (\tilde{L} + I)^{-1} \boldsymbol{q} \tag{14}$$

と表す．“$Lnnz$” を $\tilde{L}$ の非零要素数，“$Unnz$” を $\tilde{U}$ の非零要素数とすると，式 (12)，(13)，(14) を通してベクトル$\boldsymbol{g}$を得るには，以下に示す積和計算量が必要である．
(1) 前進代入：$2Lnnz$
(2) 後退代入：$2Unnz$
(3) 下三角行列とベクトルの積：$Lnnz(Lnnz - 1)$
(4) 上三角行列とベクトルの積：$Unnz(Unnz - 1)$
(5) ベクトル和：$2N$

4.2 Eisenstat 技法による計算量削減

以下では，Eisenstat 技法による式 (11) の計算量削減について記述する．式 (11) を

$$\begin{aligned}\boldsymbol{g} &= (\tilde{L} + I)^{-1} \{(\tilde{L} + I) + (\tilde{U} + I) - I\} (\tilde{U} + I)^{-1} \boldsymbol{v} \\ &= (\tilde{L} + I)^{-1} \{(\tilde{L} + I)(\tilde{U} + I)^{-1} \boldsymbol{v} + \boldsymbol{v} - (\tilde{U} + I)^{-1} \boldsymbol{v}\} \\ &= (\tilde{U} + I)^{-1} \boldsymbol{v} + (\tilde{L} + I)^{-1} \{\boldsymbol{v} - (\tilde{U} + I)^{-1} \boldsymbol{v}\}\end{aligned} \tag{15}$$

のように式変形する．次に，ベクトル$\boldsymbol{r}$，$\boldsymbol{s}$を

$$\boldsymbol{r} = (\tilde{U} + I)^{-1} \boldsymbol{v}, \tag{16}$$

$$\boldsymbol{s} = (\tilde{L} + I)^{-1} \{\boldsymbol{v} - \boldsymbol{r}\} \tag{17}$$

と定義し，ベクトル$\boldsymbol{s}$を用い，式 (15) を次のように表す．

$$\boldsymbol{g} = \boldsymbol{r} + \boldsymbol{s} \tag{18}$$

4.3 Eisenstat 技法の効果

式 (16)，(17)，(18) を通してベクトル$\boldsymbol{g}$を得るには，以下に示す積和計算量が必要である．
(1) 前進代入：$2Lnnz$
(2) 後退代入：$2Unnz$
(3) ベクトル和：$2N$

Eisenstat 技法によって，狭義の下三角行列とベクトルの積，狭義の上三角行列とベクトルの積の計算が削減された．

5. GEbE 用の計算量削減手法の提案

本節では，GEbE-GS 型前処理付き反復法の計算量削減手法を提案する．式 (8) より，両側から GEbE-GS 型前処理を適用する反復法における全体行列とベクトルの積計算を

$$\boldsymbol{h} = \prod_{k=1}^{ng}(\tilde{L_k}+I)^{-1}\tilde{A}\prod_{k=ng}^{1}(\tilde{U_k}+I)^{-1}\boldsymbol{v} \tag{19}$$

と表す．$\tilde{A}$ は対角スケーリングを適用した全体行列を各々示す．このとき，式 (3)，(7) より，$\tilde{A}$ は

$$\tilde{A} = \sum_{k=1}^{ng}\tilde{L}_k + I + \sum_{k=1}^{ng}\tilde{U}_k \tag{20}$$

と分離できる．ただし，$\tilde{D}_k$ は $\tilde{A}_k$ の対角行列とする．行列 $\tilde{M_L}^{-1}\tilde{M_U}^{-1}$ を

$$\tilde{M_L}^{-1} = \prod_{k=1}^{ng}(\tilde{L}_k+I)^{-1}, \tag{21}$$

$$\tilde{M}_{U^{-1}} = \prod_{k=ng}^{1}(\tilde{U_k}+I)^{-1} \tag{22}$$

と定義して，式 (20) を以下のように式変形する．

$$\boldsymbol{h} = \tilde{M_L}^{-1}(\sum_{k=1}^{ng}\tilde{L}_k + I + \sum_{k=1}^{ng}\tilde{U}_k)\tilde{M_U}^{-1}\boldsymbol{v}. \tag{23}$$

提案手法では，式 (23) の計算量を削減する．

5.1 行列ベクトル積の計算量

以下では，GEbE-GS 型前処理における式 (23) の計算量について記述する．ベクトル$\boldsymbol{t}$，$\boldsymbol{u}$を

$$\boldsymbol{t} = \tilde{M_U}^{-1}\boldsymbol{v}, \tag{24}$$

$$\boldsymbol{u} = (\sum_{k=1}^{ng}\tilde{L}_k + I + \sum_{k=1}^{ng}\tilde{U}_k)\boldsymbol{t} \tag{25}$$

と定義し，式 (23) を

$$\boldsymbol{h} = \tilde{M_L}^{-1}\boldsymbol{u} \tag{26}$$

と表す．式 (24)，(25)，(26) を通してベクトル$\boldsymbol{h}$を得るには，以下に示す積和計算量が必要である．

(1) 前進代入：$\sum_{e\in\mathcal{G}} N^e(N^e-1)$
(2) 後退代入：$\sum_{e\in\mathcal{G}} N^e(N^e-1)$
(3) 下三角行列とベクトルの積：$\sum_{e\in\mathcal{G}} N^e(N^e-1)$
(4) 上三角行列とベクトルの積：$\sum_{e\in\mathcal{G}} N^e(N^e-1)$
(5) ベクトル和：$2N$
(6) 必要な積和計算量の合計：$4\sum_{e\in\mathcal{G}} N^e(N^e-1)+2N$

5.2 提案手法による計算量削減

Eisenstat 技法を応用した式 (23) の計算量削減法を提案する．まず，行列 $\tilde{M}_{L1}^{-1}$，$\tilde{M}_{L2}$，$\tilde{M}_{U1}^{-1}$，$\tilde{M}_{U2}$ を

$$\tilde{M}_{L1}^{-1} = \prod_{k=1}^{ng-1}(\tilde{L}_k+I)^{-1}, \tag{27}$$

$$\tilde{M}_{L2} = \tilde{L}_{ng} + I, \tag{28}$$

$$\tilde{M}_{U1}^{-1} = \prod_{k=ng-1}^{1}(\tilde{U_k}+I)^{-1}, \tag{29}$$

$$\tilde{M}_{U2} = \tilde{U}_{ng} + I \tag{30}$$

と定義し，式 (23) を

$$\boldsymbol{h} = \tilde{M}_{L1}^{-1}\tilde{M}_{L2}^{-1}(\sum_{k=1}^{ng}\tilde{L}_k + I + \sum_{k=1}^{ng}\tilde{U}_k)\tilde{M}_{U2}^{-1}\tilde{M}_{U1}^{-1}\boldsymbol{v} \tag{31}$$

と式変形する．さらに，ベクトル$\boldsymbol{w}$を

$$\boldsymbol{w} = \tilde{M}_{U1}^{-1}\boldsymbol{v} \tag{32}$$

と定義し，式 (31) を

$$\boldsymbol{h} = \tilde{M}_{L1}^{-1}\tilde{M}_{L2}^{-1}(\sum_{k=1}^{ng}\tilde{L}_k + I + \sum_{k=1}^{ng}\tilde{U}_k)\tilde{M}_{U2}^{-1}\boldsymbol{w} \tag{33}$$

と表す．波線部分では，$\tilde{L}_{ng}$，$\tilde{U}_{ng}$ を項として持つ括弧内の行列和に対して，両側から逆行列 M_{L2}^{-1}，M_{U2}^{-1} が掛けられる．次に，Eisenstat 技法を応用して計算量の削減を考える．ベクトル$\boldsymbol{y}$，行列 B を

$$\boldsymbol{y} = \tilde{M}_{U2}^{-1}\boldsymbol{w}, \tag{34}$$

$$B = \sum_{k=1}^{ng-1}\tilde{L}_k - I + \sum_{k=1}^{ng-1}\tilde{U}_k \tag{35}$$

と定義して，式 (33) を次のように変形する．

$$\boldsymbol{h} = \tilde{M}_{L1}^{-1}\tilde{M}_{L2}^{-1}\{\underline{\tilde{M}_{L2}} + B + \underline{\tilde{M}_{U2}}\}\boldsymbol{y}. \tag{36}$$

式 (36) では，Eisenstat 技法で一般によく行われるように，式 (33) における波線部分中央の括弧内を変形して，下線部分で示すような $\tilde{M}_{L2}$，$\tilde{M}_{U2}$ という項が作れる．これらの項は，波線部分両端の逆行列 M_{L2}^{-1}，M_{U2}^{-1} との積計算によって各々単位行列化できる．その過程を以下に示す．

$$\begin{aligned}\boldsymbol{h} &= \tilde{M}_{L1}^{-1}\tilde{M}_{L2}^{-1}\{\underline{\tilde{M}_{L2}} + B + \underline{\tilde{M}_{U2}}\}\boldsymbol{y} \\ &= \tilde{M}_{L1}^{-1}\tilde{M}_{L2}^{-1}\{\tilde{M}_{L2}\boldsymbol{y} + B\boldsymbol{y} + \boldsymbol{w}\} \\ &= \tilde{M}_{L1}^{-1}\{\boldsymbol{y} + \tilde{M}_{L2}^{-1}[B\boldsymbol{y} + \boldsymbol{w}]\}.\end{aligned} \tag{37}$$

式 (37) では，$\tilde{M}_{L2}$，$\tilde{M}_{U2}$ が単位行列化された．しかし，一般的な前処理に Eisenstat 技法を適用した場合と異なり，行列 B が存在する．このとき，ベクトル$\boldsymbol{z}$，$\boldsymbol{l}$を

$$\boldsymbol{z} = B\boldsymbol{y}, \tag{38}$$

$$\boldsymbol{l} = \tilde{M}_{L2}^{-1}[\boldsymbol{z} + \boldsymbol{w}] \tag{39}$$

のように定義して，式 (37) を以下のように表す．

$$\boldsymbol{h} = \tilde{M}_{L1}^{-1}\{\boldsymbol{y} + \tilde{M}_{L2}^{-1}[\boldsymbol{z} + \boldsymbol{w}]\} = \tilde{M}_{L1}^{-1}\{\boldsymbol{y} + \boldsymbol{l}\}. \tag{40}$$

以上の手法を **GEbE-Eisenstat 技法**と呼ぶ．

5.3 GEbE-Eisenstat 技法の効果

式 (32)，(34)，(38)，(39)，(40) を通してベクトル$\boldsymbol{h}$を得

るには，以下に示す積和計算量が必要である．

(1) 前進代入：$\sum_{e\in\mathcal{G}} N^e(N^e-1)$
(2) 後退代入：$\sum_{e\in\mathcal{G}} N^e(N^e-1)$
(3) 下三角行列とベクトルの積：

$$\sum_{e\in\mathcal{G}} N^e(N^e-1) - \sum_{e\in\mathcal{G}_{ng}} N^e(N^e-1)$$

(4) 上三角行列とベクトルの積：

$$\sum_{e\in\mathcal{G}} N^e(N^e-1) - \sum_{e\in\mathcal{G}_{ng}} N^e(N^e-1)$$

(5) ベクトル和：$4N$
(6) 必要な積和計算量の合計：

$$4\sum_{e\in\mathcal{G}} N^e(N^e-1) - 2\sum_{e\in\mathcal{G}_{ng}} N^e(N^e-1) + 4N$$

5.1 節で示した計算量に比べて，狭義の下三角行列，上三角行列とベクトルの積が削減できる．ベクトル和計算は増加するが，ベクトル和計算のコストは，狭義の下三角行列，上三角行列とベクトルの積計算のコストに比べて小さいため，全体の計算量は削減できる．

6. 数 値 実 験

本節では，5 種類のテスト問題を用いた数値実験を通して，前節で示した GEbE-Eisenstat 技法の効果を評価する．比較する前処理は次の通りである．

(1) GEbEGS1：GEbE-GS 型前処理
(2) GEbEGS2：GEbE-Eisenstat 技法 + GEbE-GS1

ただし，テスト問題を解くとき，CEbE 方式と GEbE 方式を併用した．nec が 1 である場合と 2 である場合の性能も各々確かめる．

6.1 計算機環境と計算条件

数値実験で使用した計算機は Nehalem(Intel Xeon X5570，2.93GHz) を用いた．メモリは 24Gbytes，OS は RedHat Enterprise Linux 5.2 を用いた．コンパイラは Intel Fortran Compiler version 11.1，時間計測関数は C 言語の getrusage 関数を用いた．また，提案手法の計算量削減効果を正確に計測するため，コンパイラの最適化オプションは “-O0” を選択した．計算はすべて倍精度浮動小数点演算で行った．初期近似解は零とし，収束判定条件は相対残差の 2 ノルム $\|\boldsymbol{r}_{n+1}\|_2/\|\boldsymbol{r}_0\|_2$ の値が 10^{-8} 以下とした．

6.2 テスト問題

テスト問題として，正方形領域で定義された未知関数 $\boldsymbol{\phi} = \boldsymbol{\phi}(x, y)$ に対する 2 次元 Laplace 方程式を考える．

$$\frac{\partial^2\boldsymbol{\phi}}{\partial x^2} + \frac{\partial^2\boldsymbol{\phi}}{\partial y^2} = 0 \tag{41}$$

境界条件として，底辺 ($y = 0$ のとき) に $\boldsymbol{\phi}(x,\ 0) = \sin(\pi x)$，他の辺に $\boldsymbol{\phi}(x,\ y) = 0$ という Dirichlet 型境界条件を与える．領域を分割する要素は三角形要素とする．

Fig. 2 に，正方形領域におけるメッシュ分割と境界条件を示す．領域は直交座標 x-y 系を用いて，$(x, y) = [0, 1]\times[0, 1]$ で示される．さらに，三角形 1 次要素を用いて，x，y 方向に対して共に等間隔に分割した．

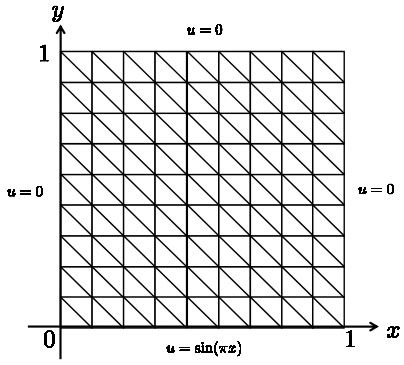

Fig. 2 An example of the computational mesh, boundary conditions and analytical domain.

Table 1 に，テスト問題の規模と削減される計算量を各々示す．“no.” は問題番号を示す．“nn” は全節点数 (number of nodes) を示す．“ne” は全要素数 (number of elements) を示す．“nc” は隣接要素群数 (number of clusters) を示す．“ratio 1” は nec が 1 の場合，削減対象となる要素数の全要素数 “ne” に対する比を示す．“ratio 2” は nec が 2 の場合，削減対象となる要素数の全要素数 “ne” に対する比を示す．どのテスト問題でも，要素集合数は，nec が 1 の場合に 8，nec が 2 の場合に 4 であった．

Table 1 Characteristics of test problems, and theoretical estimation of reduction of computational cost.

no.	nn	ne	nc	ratio 1	ratio 2
1	100^2	19,602	9,801	0.1200	0.2450
2	200^2	79,202	39,601	0.1225	0.2475
3	300^2	178,802	89,401	0.1233	0.2483
4	400^2	318,402	159,201	0.1237	0.2487
5	500^2	498,002	249,001	0.1240	0.2490
Ave.	-	-	-	0.1227	0.2477

Table 2 の観察から，行列ベクトル積の計算量を，

(1) 各隣接要素群内の要素数：nec が 1 のとき，平均約 12.3%削減でき，
(2) 各隣接要素群内の要素数：nec が 2 のとき，平均約 24.8%削減できる

ことがわかる．

6.3 実験結果

Table 2-3 に，各テスト問題における前処理効果の比較結果を示す．“no.” は問題番号を示す．“nec” は各隣接要素群内の要素数を示す．“precond.” は前処理名称を示す．“itr.” は反復回数，“total time” は合計計算時間 [単位：秒] を示す．“ave. time” は反復 1 回当たりの計算時間の平均値 [単位：ミリ秒] を示す．“$A\boldsymbol{v}$-time” は行列ベクトル積 ($A\boldsymbol{v}$) の計算時間 [単位：秒] を示す．“ratio 1” は，nec が 1 の場合の，GEbEGS1 に対する GEbEGS2 の合計計算時間の比，“ratio 2” は同平均計算時間の比，そして “ratio 3” は同行列ベクトル積計算時間の比を各々示す．

Table 2 Comparison of efficiency of two types of GEbE scheme at nec = 1.

no.	nec	precond.	itr.	total time		ave. time		$A\boldsymbol{v}$-time	
				[sec.]	ratio 1	[ms]	ratio 2	[sec.]	ratio 3
1	1	GEbEGS1	113	0.51	1.00	4.55	1.00	0.20	1.00
		GEbEGS2	113	0.50	0.98	4.41	0.97	0.18	0.88
2	1	GEbEGS1	221	4.08	1.00	18.48	1.00	1.58	1.00
		GEbEGS2	221	3.97	0.97	17.97	0.97	1.38	0.88
3	1	GEbEGS1	325	13.94	1.00	42.90	1.00	5.34	1.00
		GEbEGS2	325	13.58	0.97	41.79	0.97	4.76	0.89
4	1	GEbEGS1	427	33.36	1.00	78.13	1.00	12.85	1.00
		GEbEGS2	427	32.58	0.98	76.30	0.98	11.35	0.88
5	1	GEbEGS1	527	64.82	1.00	122.99	1.00	24.85	1.00
		GEbEGS2	527	63.12	0.97	119.77	0.97	21.84	0.88

Table 3 Comparison of efficiency of two types of GEbE scheme at nec = 2.

no.	nec	precond.	itr.	total time		ave. time		$A\boldsymbol{v}$-time	
				[sec.]	ratio 1	[ms]	ratio 2	[sec.]	ratio 3
1	2	GEbEGS1	110	0.45	1.00	4.05	1.00	0.18	1.00
		GEbEGS2	110	0.41	0.91	3.70	0.91	0.14	0.76
2	2	GEbEGS1	181	3.01	1.00	16.65	1.00	1.21	1.00
		GEbEGS2	181	2.73	0.91	15.07	0.90	0.90	0.74
3	2	GEbEGS1	238	9.49	1.00	39.86	1.00	3.81	1.00
		GEbEGS2	238	8.69	0.92	36.50	0.92	2.87	0.75
4	2	GEbEGS1	289	20.57	1.00	71.19	1.00	8.27	1.00
		GEbEGS2	289	18.66	0.91	64.56	0.91	6.09	0.74
5	2	GEbEGS1	346	38.31	1.00	110.73	1.00	15.42	1.00
		GEbEGS2	346	34.97	0.91	101.08	0.91	11.41	0.74

Table 2-3 の観察から，

(1) 各隣接要素群内の要素数：nec が 1 の場合：GEbEGS2 の合計計算時間は GEbEGS1 のおよそ 0.97 倍，同平均計算時間はおよそ 0.97 倍，そして同行列ベクトル積の計算時間はおよそ 0.88 倍である．最後の数字は **Table 1** の理論的見積もりの 12.3%と非常に近い．

(2) 各隣接要素群内の要素数：nec が 2 の場合：GEbEGS2 の合計計算時間は GEbEGS1 のおよそ 0.91 倍，同平均計算時間はおよそ 0.91 倍，そして同行列ベクトル積計算時間は GEbEGS1 のおよそ 0.75 倍である．最後の数字は **Table 1** の理論的見積もりの 24.8%と非常に近い．

ことがわかった．

7. ま と め

本論文では，GEbE-GS 型前処理つき反復法の計算量削減手法 GEbE-Eisenstat 技法を提案した．数値実験を通して，提案手法の有効性を検証した結果，提案した計算量削減手法が高速化に有効であることが分かった．さらに，提案手法の行列ベクトル積計算時間の削減効果は，理論的な見積もりと非常に近かった．

参 考 文 献

1) Hughes, T., Levit, I., Winget, J.: An element-by-element solution algorithm for problems of structural and solid mechanics, Comput. Meths. Appl. Mech. Engng., Vol.36, pp.241-254 (1983).
2) Erhel, J., Tranard, A., Vidrascu, M.: An element-by-element preconditioned conjugate gradient method on a vector computer, Para. Comput., Vol.17, pp.1051-1065 (1991).
3) Nour-Omid, B., Parlett, B. N.: Element preconditioning using splitting techniques, SIAM J. Sci. Stat. Comput., Vol.6, pp.761-770 (1985).
4) Nour-Omid, B., Parlett, B. N., Raefsky, A.: Comparision of Lanczos with conjugate gradient using element preconditioning, Proc. Domain Decompos. Symp., SIAM, pp.250-260 (1987).
5) Barragy, E., Carey, G.: A parallel element-by-element solution scheme, Int. J. Num. Meth. Engng., Vol.26, pp.2367-2382 (1988).
6) Dayde, M. J., L'exchellent, J., Gould, N. I. M.: Element-by-element preconditioners for large partially separable optimization problems, SIAM J. Sci. Stat. Comput., Vol.18, No.6, pp.1767-1787(1997).
7) Tezduyar, T. E., Liou, J.: Grouped element-by-element iteration schemes for incompressible flow computation, Comput. Phys. Comm., Vol.53, pp.441-453 (1989).
8) Liou, J., Tezduyar, T. E.: A clustered element-by-element iteration method for finite element computation, Dom. Decomp. Meth. Part. Diff. Eq., Chapter 13, SIAM, pp.140-150 (1991).
9) Chan, T. F., van der Vorst, H., A.: Approximate and Incomplete factorizations, Parallel Numerical Algorithms, ICASE/LaRC Interdisci. Seri Sci. Engng., Vol.4, pp.167-202 (1997).
10) Eisenstat, S. C.: Efficient implementation of a class of preconditioned conjugate gradient methods, SIAM J. Sci. Stat. Comput., Vol.2, pp.1-4 (1981).
11) Onoue, Y., Fujino, S.: A proposal of GS(Gauss-Seidel)-based preconditioner without extra forward and backward substitutions, The Proc. of 2nd Int. Kyoto-Forum Krylov Subspace Method, pp.29-36 (2010).