計算工学講演会論文集 Vol. 29 (2024 年 6 月)
日本計算工学会

階層型領域分割法に基づく deflation 前処理の性能評価

Performance Evaluation of Deflation Preconditioning
Based on Hierarchical Domain Decomposition

森田 直樹 [1)] 村井 拓海 [2)] 三目 直登 [3)]
Naoki Morita, Takumi Murai, and Naoto Mitsume

[1)] 博 (環境) 筑波大学システム情報系 助教（〒 305-8573 茨城県つくば市天王台 1-1-1, E-mail: nmorita@kz.tsukuba.ac.jp
[2)] 筑波大学システム情報工学研究群（〒 305-8573 茨城県つくば市天王台 1-1-1）
[3)] 博 (工) 筑波大学システム情報系 助教（〒 305-8573 茨城県つくば市天王台 1-1-1）

The Deflated Conjugate Gradient (DCG) method represents is one of preconditioned conjugate gradient method that utilize of known linearly independent vectors. In this study, we focused on the subdomain eigenmode deflation preconditioning, in which the analysis domain is divided into several regions and the lower-order eigenmodes are obtained independently in each domain and used as preconditioning. In our previous study, the method assumes that the doamin used for parallel computation and the domain where the deflation basis are acquired coincide. Thus, we proposed a preconditioning that allows the domain used for parallel computation and the domain where the deflation basis are acquired to be set independently. In this method, the analytical mesh is divided into two layers, and each layer is assigned to a domain for parallel computation and a segmentation area for preconditioning. The effectiveness of the proposed method is evaluated in terms of preconditioning and parallel computation performance using a structural analysis example with a thin plate structure.

Key Words : *Conjugate Gradient Method, Deflation, Preconditioning, Parallel computation, Domain decomposition*

1. 緒言

有限要素法は偏微分方程式の数値計算手法として有効な手法のひとつであり、構造解析など連続体を対象としたシミュレーションに広く用いられている。解析対象を詳細に評価する場合、形状そのものや物性値の分布を精緻に表現するために離散化に用いるメッシュ数が増大することで、取り扱う問題が大規模となる。大規模問題に対して、計算時間の短縮とメモリ容量の制約を緩和する目的で、分散メモリ型並列計算機を利用した並列計算手法の研究が進められてきた。

単体の計算機を用いた並列計算は、メモリ容量など, ハードウェアの制約から扱える問題が限られるため、大規模自由度問題を計算する場合には分散メモリ型並列計算機の利用が必須となる。また分散メモリ型並列計算機の利用においても、メモリ容量など個々の計算ノードのハードウェア的制約が存在する。そこで有限要素法の並列計算では、解析メッシュを複数の領域に分割して並列に計算を進める、領域分割法が広く用いられる。

本研究では、領域分割法の利用を前提に、有限要素法による構造解析を対象として、最終的に疎で正定値対称な係数行列を持つ連立一次方程式の求解に帰着する問題を扱う。このような問題では、連立一次方程式の求解を行う線形ソルバが解析に要する計算時間の多くを占める。線形ソルバのひとつである反復法は、ある手順に沿って近似解を反復的に更新し、真の解へ漸近させる手法である。この手法は高い並列計算性能の実現が期待できるものの、数値誤差の影響を受けやすく、問題によっては反復法ループが収束しない場合がある。そこで実用上、反復法の収束性・安定性を向上させるため、反復法前処理が利用される。特に並列計算を前提とした大規模問題への利用を考慮すると、並列計算性能と前処理性能の両立が課題となる。

著者らはこれまで、解くべき係数行列の低次固有モードを前処理として利用する、固有モード deflation (eigen mode deflation) に着目し、検討を進めてきた。Deflation 前処理は、既知の線形独立な複数ベクトルを係数行列から縮約する手法 [1,2] である。既知の複数ベクトルとして低次固有モードを利用する固有モード deflation では、反復法の収束性悪化の一因となる低次固有モード成分を縮約できるため、解くべき問題の条件数が低減され反復法の収束性向上が期待される。一方、固有モード deflation は条件数削減効果が期待される反面、あらかじめ固有モードが得られている場合を除き、反復開始前に固有値解析を行い、縮約に必要な低次固有モードを取得する必要がある。低次固有モードを求める固有値解析には Lanczos 法 [3] などの手法が存在するが、いずれの手法でも低次固有モードの計算コストは無視できない。

この問題を解決するため、解析領域をいくつかの領域 (subdomain) に分割し、それぞれの領域で独立に deflation 基底を取得する subdomain deflation 手法 [4] に着目し、固有モードを用いた deflation 前処理に subdomain deflation を適用した「subdomain 固有モード deflation」を考案した。この手法は、固有値解析を並列計算に利用する分割領域内の自由度に制限し、固有値問題の規模を低減することで固有値解析に必要な計算量の削減を

狙った。また、問題全体で 1 つの deflation 基底を取得した場合に比べ、subdomain deflation で得られた複数の基底を前処理に用いると、前処理性能の向上が期待される [5]。一方この研究の範囲では、並列計算に用いる分割領域と前処理に利用する deflation 基底の取得領域は一致しているとした。この場合、固有値問題の規模を低減させるには並列数を増やさなければならず、並列数が小さい実行例においては、当該前処理の有効性を示すことができていなかった。

以上より本研究では、subdomain 固有モード deflation 前処理を対象に、並列計算に用いる分割領域と前処理に利用する deflation 基底の取得領域を独立に設定可能とする前処理手法を提案する。提案手法は、解析メッシュを 2 階層に領域分割し、それぞれを並列計算に用いる分割領域と前処理適用に用いる分割領域に割り当てる。本研究の範囲では、この基礎的検討として、並列計算に用いる分割領域数よりも前処理適用に用いる分割領域数が大きい場合に注目し、検討を行う。薄板構造を有する構造解析例題を用いて、提案手法の前処理性能と並列計算性能を評価する。

2. Deflated 前処理

反復法前処理は、反復法の収束性向上を目的として、対象となる連立一次方程式にあらかじめ式変形を施す手法である。前処理は、式 (1) に示す連立一次方程式 $Ax = b$ の両辺に対し、式 (2) で示す前処理行列 M の逆行列を左から乗じる。ここで、A は大きさ $n \times n$ の正定値対称行列、x は大きさ n の解ベクトル、b は大きさ n の右辺ベクトルである。このような前処理を特に左前処理と呼ぶ。前処理付き反復法の計算時間は式 (2) の求解時間と前処理行列 M の生成時間の合計で定まる。

$$Ax = b \tag{1}$$

$$M^{-1}Ax = M^{-1}b \tag{2}$$

Deflation 前処理とは、事前に取得した既知の基底の組を deflation 基底として入力し、解くべき係数行列から入力基底の情報を除外することで、収束性の向上を狙う手法である。連立一次方程式 $Ax = b$ の係数行列 A に対して、入力する n_{def} 本 ($n_{\mathrm{def}} \ll n$) の deflation 基底ベクトル w_1 から $w_{n_{\mathrm{def}}}$ を式 (3) のように並べて $n \times n_{\mathrm{def}}$ の行列 W を構成する。

$$W = \begin{bmatrix} w_1 & \ldots & w_{n_{\mathrm{def}}} \end{bmatrix} \tag{3}$$

ここで大きさ n_{def} かつ 0 でないベクトル y を使用し、解 x を次のように表す。

$$x = Wy + (x - Wy) \tag{4}$$

A 直交の定義より、式 (5) が得られる。ここで (a, b) はベクトル a、b の内積を示す。続けて式 (5) を展開することで式 (6) を得る。ここで W^t は W の転置を表す.

$$(W, Ax - AWy) = 0 \tag{5}$$

$$W^t Ax = W^t AWy \tag{6}$$

式 (6) を整理し、左から W を乗じて式 (7) を得る。ここで $n \times n$ の行列 $Q = W(W^t AW)^{-1} W^t$ を定義し、式 (8) で表す。

$$Wy = W(W^t AW)^{-1} W^t Ax \tag{7}$$

$$Wy = QAx \tag{8}$$

さらに $n \times n$ の行列 $P = I - AQ$ を定義して、式 (8) を式 (4) に代入し整理すると、解ベクトル x は式 (9) で得られる。ここで I は単位行列である。

$$x = Qb + P^t x \tag{9}$$

ここで式 (9) 右辺第 1 項は、既知の行列 Qb から求まる。次に式 (9) 右辺第 2 項は、$AP^t = PA$ の関係を用いて、式 (10) から得られる。

$$PAx = Pb \tag{10}$$

式 (10) より、deflation は前処理行列 M_{def}^{-1} を用いて式 (11) で表される。

$$M_{\mathrm{def}}^{-1} = I - AW(W^t AW)^{-1} W^t \tag{11}$$

Deflation 前処理では、前処理後の行列は対称性を維持するが特異となる。このような行列に対しても、共役勾配法を用いた求解が可能であることが知られており、本研究ではこれらの先行研究に基づき、前処理適用後の連立一次方程式求解についても共役勾配法を利用した。

以上の関係に基づき前処理つき共役勾配法を構成する。Deflation 前処理を適用した共役勾配法は deflated 共役勾配法と呼ばれる場合がある。なお、deflated 共役勾配法には標準的な前処理 $\overline{M}$ を併用可能である [6]。前処理付き deflated 共役勾配法のアルゴリズムを Algorithm 1 に示す。

Algorithm 1 Deflated Preconditioned Conjugate Gradient Method

function DEFLATED_CG$(A, b, x, W, \overline{M})$
  $r_0 \leftarrow b - Ax_0$
  $z_0 \leftarrow \overline{M}^{-1} r_0$
  $p_0 \leftarrow z_0$
  for $(i = 1, 2, 3, \ldots)$ **do**
    $w_{i-1} \leftarrow PAp_{i-1}$
    $\alpha_i \leftarrow (r_{i-1}^t z_{i-1})/(p_{i-1}^t w_{i-1})$
    $x_i \leftarrow x_{i-1} + \alpha_i p_{i-1}$
    $r_i \leftarrow r_{i-1} - \alpha_i w_{i-1}$
    if $(\|r_i\|/\|r_0\| < \varepsilon)$ **then**
      exit
    end if
    $z_i \leftarrow \overline{M}^{-1} r_i$
    $\beta_i \leftarrow \frac{r_i^t z_i}{r_{i-1}^t z_{i-1}}$
    $p_i \leftarrow z_i + \beta_i p_{i-1}$
  end for
  return $x_i = Qb + P^t x_i$
end function

3. 階層型領域分割に基づく deflated 前処理

4. Overlapping 型領域分割法

本研究では、有限要素離散化で得られる係数行列の非ゼロ構造に対応するグラフ構造を取得し、グラフ分割の結果を元にしてデータ分割を行うとの手順をとる。有限要素法の場合、ある要素を構成する節点 a と 節点 b は相互作用をもつ。ここで離散化して得られる連立一次方程式の係数行列は、第 (a,b) 番目の行列要素が非零となる。この関係をノード v_a、v_b とエッジ $e_k = \{v_a, v_b\}$ として、全ての節点・要素を変換することでグラフ G を得る。

係数行列の非零構造に対応するグラフ構造を得られたあとは、領域分割をグラフの最小 k-way カット問題とみなしてデータ分割する。グラフの最小 k-way カット問題は、グラフ G に対し、ノード集合 V を p 個の部分集合に分割することで与えられる。このとき、各部分集合 $V^{(i)}$ $(i = 1,\dots,p)$ のノード数は均一、かつ異なる部分集合に跨るエッジ数を最小にする。ここで $V^{(i)}$ は Eq. (12)、Eq. (13) を満たす。

$$V^{(i)} \cap V^{(j)} = \emptyset \ (i \neq j) \tag{12}$$

$$\bigcup_{i=1}^{p} V^{(i)} = V \tag{13}$$

グラフ分割によって得られたノードの部分集合 $V^{(i)}$ は、他の領域と重複なく分けられている。グラフ分割される前の単一グラフ構造と比較すると、本来のエッジ情報が欠けているため、数値シミュレーションに必要な情報が不完全な状態となっている。そこで各ノードに注目したとき、グラフ分割がなされる前の単一グラフ構造と同等の相互作用関係情報を有するよう、不足したグラフ情報を補うオーバーラップ領域を定める。

オーバーラップ領域を加味したノードの部分集合 $\overline{V}^{(i)}$ $(i = 1,\dots,p)$ は、Eq. (14) で与えられる。

$$\overline{V}^{(i)} = V^{(i)} \cup V^{(i)}_{\mathrm{ovl}} \tag{14}$$

ここで、$V^{(i)}_{\mathrm{ovl}}$ はグラフ G のノード集合 $V^{(i)}$ 以外に属するノードのうち、ノード集合 $V^{(i)}$ に隣接するノードの集合である。ノード集合 $\overline{V}^{(i)}$ において、集合 $V^{(i)}$ を領域 i の内部領域と呼び、集合 $V^{(i)}_{\mathrm{ovl}}$ を領域 i のオーバーラップ領域と呼ぶ。さらに分割前のグラフ G のうち、ノード集合 $\overline{V}^{(i)}$ で構成されるエッジ集合 $\overline{E}^{(i)}$ を定義して、部分グラフは $G^{(i)} = (\overline{V}^{(i)}, \overline{E}^{(i)})$ で与えられる。

5. 階層型領域分割に基づく前処理計算

グラフ G の分割により分割グラフ $G^{(i)}$ が得られたとき、領域 i の内部領域 $V^{(i)}$ に対応するベクトル $\boldsymbol{x}^{(i)}$ を Eq. (15)、領域 i の内部領域 $V^{(i)}$ が領域 j の内部領域 $V^{(j)}$ から受ける作用を表す行列 $\boldsymbol{A}^{(i,j)}$ を Eq. (16) で表す。

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} \boldsymbol{x}^{(i)}_1 \\ \boldsymbol{x}^{(i)}_2 \\ \vdots \\ \boldsymbol{x}^{(i)}_{|V^{(i)}|} \end{bmatrix} \tag{15}$$

$$\boldsymbol{A}^{(i,j)} = \begin{bmatrix} \boldsymbol{A}^{(i,j)}_{1,1} & \boldsymbol{A}^{(i,j)}_{1,2} & \dots & \boldsymbol{A}^{(i,j)}_{1,|V^{(j)}|} \\ \boldsymbol{A}^{(i,j)}_{2,1} & \boldsymbol{A}^{(i,j)}_{2,2} & \dots & \boldsymbol{A}^{(i,j)}_{1,|V^{(j)}|} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{(i,j)}_{|V^{(i)}|,1} & \boldsymbol{A}^{(i,j)}_{|V^{(i)}|,2} & \dots & \boldsymbol{A}^{(i,j)}_{|V^{(i)}|,|V^{(j)}|} \end{bmatrix} \tag{16}$$

ここで、$|V|$ はグラフのノード数（= 内部領域に含まれる節点数）を示す。また領域 i と領域 j が隣接していない場合、ブロック行列 $\boldsymbol{A}^{(i,j)}$ は零行列となる。

また、ローカル計算点番号–グローバル計算点番号の対応関係を示す大きさ $|V| \times |V^{(k)}|$ の 0-1 行列 $\boldsymbol{P}^{(k)}$ を用いてデータ分割前のベクトル $\boldsymbol{x}$、行列 $\boldsymbol{A}$ は、それぞれ Eq. (17)、Eq. (18) で表される。

$$\boldsymbol{x} = \sum_{k=1}^{p} \boldsymbol{P}^{(k)} \boldsymbol{x}^{(k)} \tag{17}$$

$$\boldsymbol{A} = \sum_{k=1}^{p} \sum_{l=1}^{p} \boldsymbol{P}^{(k)} \boldsymbol{A}^{(k,l)} \, {}^{t}\boldsymbol{P}^{(l)} \tag{18}$$

ここで、左上添字 t は転置記号である。

本研究における階層型領域分割は、第 1 階層目のグラフ分割により領域 i の内部節点集合 $V^{(i)}$ に関する行列 $\boldsymbol{A}^{(i,i)}$ が得られた後、行列 $\boldsymbol{A}^{(i,i)}$ の非ゼロ構造に対応するグラフ $H^{(i)}$ を得る。次に、第 2 階層目のグラフ分割として領域 i のグラフ $H^{(i)}$ を分割数 q でグラフ分割する。この操作により、第 1 階層目の領域 i、第 2 階層目の領域 j に対応するブロック行列 $\boldsymbol{A}^{(i,i,j,j)}$ を得る。このように得られたブロック行列に対し、行列 $\boldsymbol{A}^{(i,i,j,j)}$ $(1 \leq i \leq p,\ 1 \leq j \leq q)$ に対し、低次固有ベクトルを取得して、deflation 前処理の既知の基底を計算する。

ここで、階層型領域分割を考慮した subdomain deflation によって得られる大きさ $n \times (n_{\mathrm{def}}\, p\, q)$ の基底 W は、第 1 階層目の領域 i、第 2 階層目の領域 j で得られる大きさ $n^{(i,j)} \times n_{\mathrm{def}}$ のブロック行列 $W^{(i,j)}$ によって、式 (19) で得られる。本研究の範囲内では、1 領域あたりの基底数 n_{def} は分割領域によらず一定とした。

$$W = \begin{bmatrix} W^{(1,1)} & 0 & \dots & 0 \\ 0 & W^{(1,2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W^{(p,q)} \end{bmatrix} \tag{19}$$

6. 計算性能評価

数値例として、薄板構造を有する構造解析例題を用いて、提案手法の前処理性能と並列計算性能を評価する。大阪大学 Squid スーパーコンピュータ [7] を利用した分散メモリ型並列環境において、提案手法の前処理性能と並列計算性能を測定し、その有効性を評価する。なお、本研究の研究開発基盤として monolis を用いた [8]。グラフ分割については、グラフ分割ライブラリ METIS [9] により、計算負荷の均一化および通信量の最小化に基づく領域分割を行った。

7. 結言

本研究では、解析領域をいくつかの領域に分割し、それぞれの領域で独立に低次固有モードを取得、前処理として利用する subdomain 固有モード deflation 手法に着目した。従来、並列計算に用いる分割領域と前処理に利用する deflation 基底の取得領域は一致しているとした方法に対し、並列計算に用いる分割領域と前処理に利用する deflation 基底の取得領域を独立に設定可能とする前処理手法を提案した。この方法は、解析メッシュを 2 階層に領域分割し、それぞれを並列計算に用いる分割領域と前処理適用に用いる分割領域に割り当てる。

提案手法の有効性は、薄板構造を有する構造解析例題を用いて、提案手法の前処理性能と並列計算性能を評価する。詳細な内容については、口頭発表で紹介する。

謝辞: 本研究は、JSPS 科研費 23K16891、JSPS 科研費 23K24857、JST 創発的研究支援事業 JPMJFR215S の助成および学際大規模情報基盤共同利用・共同研究拠点 (課題番号:jh240017) の支援を受けたものである。ここに謝意を表す。

参考文献

[1] Nicolaides, R. A., Deflation of conjugate gradients with applications to boundary value problems, *SIAM Journal on Numerical Analysis*, **24**-2, 1987, pp. 355–365.

[2] Tang, J. M., MacLachlan, S. P., Nabben, R., and Vuik, C., A comparison of two-level preconditioners based on multigrid and deflation, *SIAM Journal on Matrix Analysis and Applications*, **31**-4, 2010, pp. 1715–1739.

[3] Golub, G. H. and Underwood, R., The block lanczos method for computing eigenvalues, *Mathematical software*, 1977, pp. 361–377.

[4] Frank, J. and Vuik, C., On the construction of deflation-based preconditioners, *SIAM Journal on Scientific Computing*, **23**-2, 2001, pp. 442–462.

[5] Agathos, K., Dodwell, T., Chatzi, E., and Bordas, S. P. A., An adapted deflated conjugate gradient solver for robust extended/generalised finite element solutions of large scale, 3D crack propagation problems, *Computer Methods in Applied Mechanics and Engineering*, **395**, 2022, p. 114937.

[6] Saad, Y., Yeung, M., Erhel, J., and Guyomarc'h, F., A deflated version of the conjugate gradient algorithm, *SIAM Journal on Scientific Computing*, **21**-5, 2000, pp. 1909–1926.

[7] Cybermedia center blog archive squid, osaka university, `http://www.hpc.cmc.osaka-u.ac.jp/squid/`, 2023 年 12 月 6 日閲覧.

[8] monolis: monolithic linear solver based on domain decomposition, `https://www.kz.tsukuba.ac.jp/~nmorita/monolis.html`, 2023 年 12 月 22 日閲覧.

[9] Karypis, G. and Kumar, V., Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, *Computer Science & Engineering (CS&E) Technical Reports*, **97**-061, 1997.