

# 特異値分解の逐次更新手法を利用した deflation 前処理の性能評価

Performance Evaluation of Deflation Preconditioners  
Using Incremental Singular Value Decomposition

清水嶺<sup>1)</sup> 三目直登<sup>2)</sup> 森田直樹<sup>3)</sup>

Rei Shimizu, Naoto Mitsume, and Naoki Morita

<sup>1)</sup>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, E-mail: s2011207@tsukuba.ac.jp)

<sup>2)</sup>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, E-mail: mitsume@kz.tsukuba.ac.jp)

<sup>3)</sup>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)

Deflation is one of the preconditioner to accelerate the convergence of conjugate gradient (CG) method. Deflation uses the linearly independent bases (deflation bases) which include the information of a target system. In this paper, we consider the subdomain incremental proper orthogonal decomposition modes as the deflation bases. We apply the deflation to the dynamic structure problem, and compare the preconditioned deflated CG method to the standard preconditioned CG method in terms of the number of iterations and the calculation time assuming that we use the domain decomposition method.

**Key Words :** Dynamic Structural Analysis, Conjugate Gradient Method, Deflation Preconditioning, Incremental Singular Value Decomposition

## 1. 序論

有限要素法は偏微分方程式の数値解析手法の 1 つである。精緻な解析をするためには離散化の際メッシュを細かくする必要があり、離散化後の問題は次元が大きい大規模問題となる。大規模問題を解くためには、計算時間とメモリ使用量の観点から分散メモリ型並列計算機を使用した並列計算が必要となる。有限要素法の並列計算手法として領域分割法が広く使用されており、本研究ではオーバーラッピング型領域分割法に基づく並列有限要素法による動的構造解析を対象とする。

本研究では、有限要素法による離散化の結果正定値対称な係数行列を持つ連立一次方程式に帰着する問題を扱う。連立一次方程式の求解は計算時間のボトルネックとなっている。係数行列が正定値対称な大規模問題では、求解手法として conjugate gradient (CG) 法が一般に使用される。CG 法は反復計算によって近似解を求める手法であり、問題によって収束までに多くの反復回数を必要とするため、実用上収束性の向上が必要である。収束性向上の手段として前処理が広く使用される。前処理は CG 法の収束性が向上するように連立一次方程式を変形するものである。既存手法として、対角スケーリングという係数行列の対角成分を利用する前処理があるが、収束性向上には限界がある。

本研究では、既知の基底を用いる前処理である deflation について検討する。Deflation は対角スケーリングや修正コレスキー分解等の前処理と併用が可能であり、併用することにより計算時間を削減できることが示されている [1,3,8]。本研究では対角スケーリングを使用する。Deflation に用いる基底を deflation 基底と呼び、本研究で対象とする動解析において、proper orthogonal

decomposition (POD) 基底を deflation 基底に用いる POD deflation が提案されている [2]。これは、解析結果 (スナップショット) を並べた行列を特異値分解して得られる左特異ベクトルを deflation 基底に用いる手法であり、線形動解析において有効であることが示されている [3]。一方境界条件が変化したり非線形性が生じたりする問題では、前処理性能維持のために POD 基底の再計算が必要と考えられるが、POD の再計算による計算時間の増加とスナップショットの保持による使用メモリ量の増加という課題がある。ここで前処理性能という言葉は、前処理によって CG 法の収束までの反復回数をどれだけ削減できたかという意味で用いている。

本研究ではこの課題解決のために、特異値分解の逐次更新手法である incremental singular value decomposition (SVD) [4] によって POD 基底を更新する incremental POD deflation [5] について検討する。Incremental POD deflation の構造解析への適用例は確認できていないため、本研究で有効性を評価する。また、解析領域を分割し各領域内の節点の情報のみを保持した deflation 基底を求める subdomain deflation が提案されている [6]。Subdomain deflation の導入によって deflation の前処理性能が向上することが示されている [7,8]。本研究では、subdomain deflation と incremental POD deflation を組み合わせた subdomain incremental POD deflation を提案し、その前処理性能と計算時間から有効性を評価する。本研究では基礎的検討として線形動解析を対象とする。

以降の各章において、大文字細字は行列、小文字太字はベクトル、小文字細字はスカラーを表す。また、行列は全て実行列、ベクトルは全て実数ベクトル、スカラーは全て実数とする。

## 2. Deflated CG 法

Deflated CG 法には複数のアルゴリズムが存在するが、比較的計算コストが低くロバストなアルゴリズム [9] を採用する [1]。

連立一次方程式を式 (1) のように表す。

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

ここで  $A$  は正定値対称行列、 $x$  は解ベクトル、 $b$  は右辺ベクトルとする。

$n$  本の基底  $w_1 \dots w_n$  で構成される行列  $W$  を式 (2) のように導入する。

$$W = [w_1 \dots w_n] \quad (2)$$

この基底  $w_1 \dots w_n$  を deflation 基底と呼ぶ。

$x$  は、 $W$  と、零ベクトルでないベクトル  $y$  を用いて式 (3) のように表せる。

$$x = Wy + (x - Wy) \quad (3)$$

CG 法によって  $x$  を求める場合、 $W$  の各列と  $x - Wy$  は  $A$  直交でなければならないから、式 (4) が成り立つ。

$$\begin{aligned} W^T A(x - Wy) &= 0 \\ W^T A W y &= W^T A x \end{aligned} \quad (4)$$

ここで  $W^T$  は  $W$  の転置を表す。

式 (4) を  $Wy$  について整理すると、 $Wy$  と  $x - Wy$  はそれぞれ式 (5)、式 (6) となる。

$$\begin{aligned} Wy &= W(W^T A W)^{-1} W^T A x \\ &= W(W^T A W)^{-1} W^T b \end{aligned} \quad (5)$$

$$x - Wy = (I - W(W^T A W)^{-1} W^T A)x \quad (6)$$

ここで  $(W^T A W)^{-1}$  は  $W^T A W$  の逆行列を表す。

ここで射影  $P$  を式 (7) で定義する。

$$P \equiv I - AW(W^T A W)^{-1} W^T \quad (7)$$

式 (7) を式 (5) と式 (6) に代入すると、 $Wy$  と  $x - Wy$  はそれぞれ式 (8) と式 (9) となる。

$$Wy = (I - P^T)x \quad (8)$$

$$x - Wy = P^T x \quad (9)$$

式 (8) と式 (9) を式 (3) に代入すると、 $x$  は式 (10) となる。

$$x = (I - P^T)x + P^T x \quad (10)$$

式 (10) の第一項は式 (5) より求まる。

式 (10) の第二項について、まず射影  $P^T$  について式 (11) の関係が成り立つ。

$$\begin{aligned} AP^T &= A(I - W(W^T A W)^{-1} W^T A) \\ &= (I - AW(W^T A W)^{-1} W^T)A \\ &= P^T A \end{aligned} \quad (11)$$

### Algorithm 1 対角スケーリング付き deflated CG 法

```

1:  $\hat{r}_0 = b - Ax_0$ 
2:  $r_0 = P\hat{r}_0$ 
3:  $y_0 = M^{-1}r_0$ 
4:  $p_0 = y_0$ 
5: for  $i = 0, 1, \dots$  do
6:    $z_i = PAp_i$ 
7:    $\alpha_i = \frac{r_i^T y_i}{p_i^T z_i}$ 
8:    $x_{i+1} = x_i + \alpha_i p_i$ 
9:    $r_{i+1} = r_i - \alpha_i z_i$ 
10:  if  $\frac{\|r_i\|_2}{\|r_0\|_2} \leq \varepsilon$  then
11:    exit
12:   $y_{i+1} = Mr_i$ 
13:   $\beta_i = \frac{r_{i+1}^T y_{i+1}}{r_i^T y_i}$ 
14:   $p_{i+1} = y_i + \beta_i p_i$ 
15:  $x = P^T x_i + Qb$ 

```

式 (11) より式 (12) が導かれる。

$$\begin{aligned} AP^T x &= PAx = Pb \\ PAx &= Pb \end{aligned} \quad (12)$$

式 (12) は係数行列が対称半正定値行列となり CG 法で解ける [7,9]。式 (12) を CG 法によって解き、式 (10) によって式 (1) の解を求めるのが deflated CG 法である。

Deflation を対角スケーリングと併用する場合、その前処理行列を  $M$  をすると、式 (12) の代わりに式 (13) を CG 法によって解く。

$$M^{-1}PAx = M^{-1}Pb \quad (13)$$

Algorithm 1 に対角スケーリング付き deflated CG 法のアルゴリズムを示す。ここで  $r$  は残差ベクトル、 $p$  は探索方向ベクトル、 $\alpha, \beta$  は探索幅を決めるスカラー値、 $y, z$  は計算結果を一時的に保持するためのベクトル、 $\varepsilon$  は収束判定閾値、右下添え字  $i$  は反復計算のインデックス、 $\|\cdot\|$  は L2 ノルムを表す。Algorithm 1 において、1 ~4 行目は初期値の設定、5~14 行目は CG 法の反復計算、15 行目は式 (10) の計算を行っている。Algorithm 1 の 6 行目の  $P$  は、 $(W^T A W)^{-1}$  の逆行列を陽に計算するのではなく式 (14) を  $\mu$  について解くことで求める。

$$W^T A W \mu = W^T A p \quad (14)$$

## 3. Subdomain incremental POD deflation

### (1) Incremental POD

スナップショットを並べた  $m \times n$  行列を  $X$  とする。 $X$  は式 (15) のように特異値分解できる。

$$X = USV^T \quad (15)$$

ここで  $U$  と  $V$  は直交行列、 $S$  は対角行列とする。 $U$  の各列を左特異ベクトル、 $S$  の対角成分を特異値、 $V$  の各列を右特異ベクトルという。

式(15)のうち、特異値の大きい順に  $r$  個の特異値とそれに対応する左右特異ベクトルのみ取り出すと、 $X$  は式(16)のように表せる。

$$X \approx U_r S_r V_r^T \quad (16)$$

ここで  $U_r$  と  $V_r$  は直交行列、 $S_r$  は対角行列である。 $U_r$  の各列を deflation 基底として使うのが POD deflation である。

$X$  が別の行列  $X'$  に変化したとき、 $X'$  の特異値分解を式(16)を利用して求めるのが incremental SVD である。 $X'$  が  $X$  とある行列  $A, B$  を用いて式(17)のように表されるとする。

$$X' = X + AB^T \quad (17)$$

式(16)を式(17)に代入して式変形すると、 $X'$  は式(18)となる。

$$\begin{aligned} X' &\approx U_r S_r V_r^T + AB^T \\ &= [U_r \ A] \begin{bmatrix} S_r & O \\ O & I \end{bmatrix} [V_r \ B]^T \\ &= [U_r \ U_r U_r^T A + (I - U_r U_r^T) A] \begin{bmatrix} S_r & O \\ O & I \end{bmatrix} \\ &\quad [V_r \ V_r V_r^T B + (I - V_r V_r^T) B]^T \end{aligned} \quad (18)$$

式(18)において、 $(I - U_r U_r^T)A$  と  $(I - V_r V_r^T)B$  をそれぞれ QR 分解するとそれぞれ式(19)、式(20)のように表せる。

$$(I - U_r U_r^T)A = Q_A R_A \quad (19)$$

$$(I - V_r V_r^T)B = Q_B R_B \quad (20)$$

ここで  $Q_A$  と  $Q_B$  は直交行列、 $R_A$  と  $R_B$  は上三角行列とする。

式(19)と式(20)を式(18)に代入すると式(21)となる。

$$\begin{aligned} X' &\approx [U_r \ Q_A] \begin{bmatrix} I & U_r^T A \\ O & R_A \end{bmatrix} \begin{bmatrix} S_r & O \\ O & I \end{bmatrix} \begin{bmatrix} I & V_r^T B \\ O & R_B \end{bmatrix}^T \\ &\quad [V_r \ Q_B]^T \\ &= [U_r \ Q_A] \left( \begin{bmatrix} S_r & O \\ O & O \end{bmatrix} + \begin{bmatrix} U_r^T A \\ R_A \end{bmatrix} \begin{bmatrix} V_r^T B \\ R_B \end{bmatrix}^T \right) \\ &\quad [V_r \ Q_B]^T \\ &= [U_r \ Q_A] K [V_r \ Q_B]^T \end{aligned} \quad (21)$$

式(21)において行列  $K$  を式(22)で定義した。

$$K = \begin{bmatrix} S_r & O \\ O & O \end{bmatrix} + \begin{bmatrix} U_r^T A \\ R_A \end{bmatrix} \begin{bmatrix} V_r^T B \\ R_B \end{bmatrix}^T \quad (22)$$

$K$  を特異値分解すると式(23)となる。

$$K = U_K S_K V_K^T \quad (23)$$

ここで  $U_K$  と  $V_K$  は直交行列、 $S_K$  は対角行列とする。

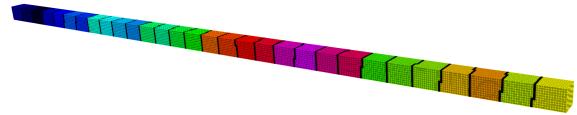


図-1 解析モデル

式(23)を式(21)に代入すると式(24)となり、 $X'$  の特異値分解が得られる。

$$X' \approx ([U_r \ Q_A] U_K) S_K ([V_r \ Q_B] V_K)^T \quad (24)$$

$[U_r \ Q_A] U'$  を deflation 基底として使うのが incremental POD deflation である。

Incremental POD の利点は 2 つある [4]。1 つは計算量の削減である。 $X'$  を直接特異値分解する場合の計算量は  $O(mn^2)$  だが、incremental SVD では新たに  $c$  本のスナップショットを追加した特異値分解の計算量は  $O(m(r+c)^2)$  となる。そのため  $r+c < n$  の場合に incremental POD は POD より計算量が小さい。もう 1 つは、 $X'$  の特異値分解を求めた後  $X'$  そのものを破棄してもよいためメモリ使用量を削減できることである。

## (2) 提案手法

本研究では、subdomain deflation と incremental POD deflation を組み合わせた subdomain incremental POD deflation を提案する。提案手法では、領域ごとに独立して incremental SVD によって POD 基底を更新する。

## 4. 性能評価

### (1) 解析条件

線形動的構造解析において、従来手法である対角スケーリング付き CG 法と、提案手法である subdomain incremental POD を利用した対角スケーリング付き deflated CG 法の、前処理性能と解析全体の計算時間を比較した。incremental POD と POD の比較のため、POD 基底を用いた deflation の結果も取得した。

解析モデルは片持ち梁とする。図 1 に領域分割数 30 の場合の解析メッシュを示す。荷重条件として周期的に振幅と方向が変化する荷重を与えた。解析の全タイムステップは 600 とし、荷重方向を 200 ステップごとに  $x$ 、 $y$ 、 $z$  方向と変化させた。Deflation 基底は、150 ステップ毎に POD 基底、300、400 ステップ毎に incremental POD 基底を取得した。POD と incremental POD において、最大特異値との相対値が  $10^{-8}$  未満の特異値に対応する特異ベクトルは除外した。今回の検討では、並列計算において各プロセスが担当する領域と subdomain deflation における各領域を一致させて、ともに 1(逐次)、30 と変化させた。六面体一次要素を使用し自由度数は 145563 とした。物性値はヤング率を 206GPa、ポアソン比を 0.3 とした。直接積分法として Newmark-beta 法、減衰項に Rayleigh 減衰を使用した。CG 法と deflated CG 法の収束判定閾値は  $10^{-8}$ 、式(14)を CG 法で解く際の収束判定閾値は  $10^{-10}$  とした。計算機は Linux サーバー (CPU:Intel Xeon Gold 6248、メモリ:786 GB、OS:Ubuntu 20.04.3 LTS) を使用した。

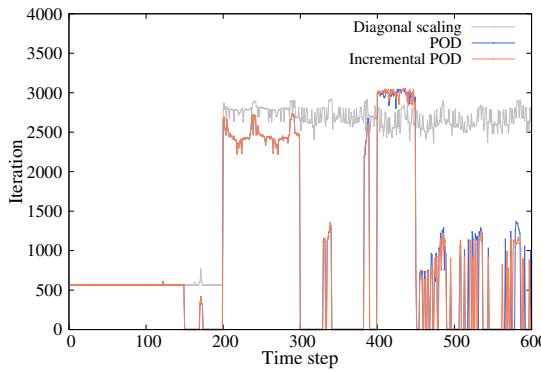


図-2 タイムステップ毎の CG 法の反復回数 (逐次)

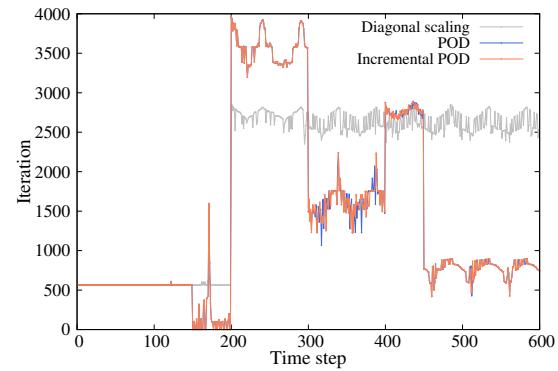


図-3 タイムステップ毎の CG 法の反復回数 (30 並列)

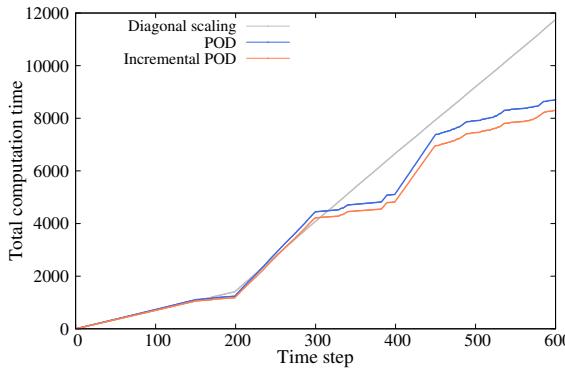


図-4 タイムステップ毎の累積計算時間 (逐次)

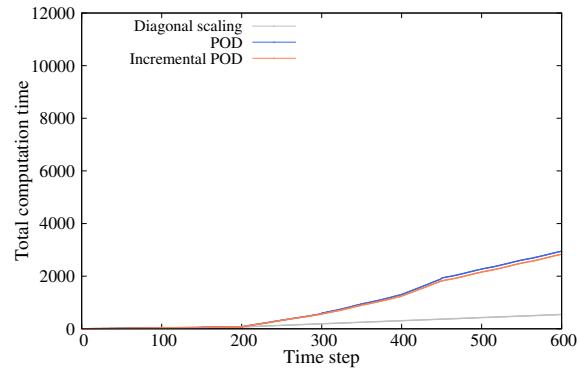


図-5 タイムステップ毎の累積計算時間 (30 並列)

## (2) 結果

図 2、図 3 にそれぞれ逐次、30 並列におけるタイムステップ毎の CG 法の反復回数を示す。ここで凡例は、“Diagonal scaling”は従来手法、“Incremental POD”は提案手法、“POD”は POD 基底を用いた deflated CG 法を表す。

図 2 と図 3 より、逐次と 30 並列ともに、提案手法は従来手法よりも前処理性能が高いことがわかる。また図 2 と図 3 を比較すると、subdomain deflation によって前処理性能は向上しなかったといえる。これは、解析モデルが単純形状であったため、局所的に値をもつ基底よりも大域的に値を持つ基底の方が deflation 基底に適していたためと考えている。また図 2 と図 3 より、incremental POD と POD ではほぼ同じ前処理性能が得られることがわかる。Incremental SVD は SVD を近似的に計算するものであるが、十分な精度で近似できているといえる。

図 4、図 5 にそれぞれ逐次、30 並列におけるタイムステップ毎の累積計算時間を示す。ここで凡例は図 2、図 3 と同じである。

図 4 と図 5 より、逐次では提案手法は従来手法よりも累積計算時間を削減できたが、30 並列では逆に増大してしまった。Incremental POD の場合、逐次では 150、300、450 ステップ目で取得した deflation 基底数はそれぞれ 7、13、24 だった。これに対し 30 並列では、1 領域あたりの deflation 基底数はそれぞれ 2、5、8 であり、全体ではそれぞれ 60、150、240 だった。そのため、領域分割数の増加に伴い deflation 基底数が増加したこと

によって deflation にかかる計算時間も増加したのに対し、領域分割数が増加しても deflated CG 法の反復回数は削減されなかったため、解析全体の計算時間は増えてしまったと考えられる。また 30 並列の場合、300、450 ステップ目において、POD にかかった時間はそれぞれ 16.8 秒、28.6 秒だった。これに対し incremental POD にかかった時間それぞれ 1.89 秒、2.92 秒だった。したがって、incremental POD は POD より計算時間が小さいといえる。

## 5. 結論と今後の展望

本研究では、特異値分解の逐次更新手法を利用した deflation について検討を行った。提案手法である subdomain incremental POD deflation を線形動的構造解析に適用し、前処理性能と計算時間を評価した。数値実験により、逐次計算では提案手法の有効性が認められたが並列計算では認められなかった。また、POD を incremental POD に置き換えることにより、前処理性能を維持したまま計算時間を削減できることが示された。

今後の課題として、領域分割数と並列数の独立な設定による前処理性能と計算時間の両立を図ることが挙げられる。また今後の展望として非線形解析への適用が挙げられる。

**謝辞:** 本研究は、JST 創発的研究支援事業 JP-MJFR215S、JSPS 科研費 23K16891、学際大規模情報基盤共同利用・共同研究拠点 (JHPCN) jh240017 の助

成を受けたものである。ここに記して謝意を表する。

### 参考文献

- [1] J.M. Tang · R. Nabben · C. Vuik · Y.A. Erlangga, Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods, Journal of Scientific Computing, 2009, Vol.39, p.340-370.
- [2] G.B. Diaz Cortes, C. Vuik, J.D. Jansen, On POD-based deflation vectors for DPCG applied to porous media problems, Journal of Computational and Applied Mathematics, 2017, Vol.330, pp. 193-213.
- [3] 平野皓大, 三目直登, 森田直樹. Snapshot PODに基づく縮約基底を用いた deflation 前処理の性能評価, 日本計算工学会第 29 回計算工学講演会, 2022.
- [4] Matthew Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra and its Applications, Vol. 415, 2006, p20-30.
- [5] Youngsoo Choi, Geoffrey Oxberry, Daniel White, Trenton Kirchdoerfer, Accelerating design optimization using reduced order models. arXiv:1909.11320, <https://arxiv.org/abs/1909.11320>, 2019.
- [6] Nicolaides, R. A., Deflation of conjugate gradients with applications to boundary value problems, SIAM Journal on Numerical Analysis, 24-2, 1987, pp. 355-365.
- [7] Frank, J. and Vuik, C., On the construction of deflation-based preconditioners, SIAM Journal on Scientific Computing, 23-2, 2001, pp. 442-462.
- [8] 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.
- [9] C. Vuik, A. Segal, J. A. Meijerink, An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients, Journal of Computational Physics 1999, Vol.152, p.385-403.