Vol.2012-HPC-134 No.2
2012/6/1

適切な block-size の決定法を用いた Block Gram-Schmidt 法の並列化

松 尾 洋 一[†1] 野 寺 隆[†2]

Parallel Block Gram-Schmidt 法による QR 分解は，行列 X を m 列のブロックごとに直交化することで，QR 分解を高速に行うことができる．しかし，行列 X によって，適切な block-size は様々である．最適な block-size を求めるために何度も QR 分解を行うことは非効率である．本発表では，適切な block-size を Parallel Block Gram-Schmidt 法の実行中に自動的に決定する手法について提案する．そして，数値実験によりこの手法が有効であることを示す．

The parallelization for the Block Gram-Schmidt Orthogonalization with the Optimal block-size

Yoichi Matsuo[†1] and Takashi Nodera[†2]

We propose an adaptive procedure for determining the optimal block-size for the Parallel Block Gram-Schmidt orthogonalization (PBGS), which is a process for computing the QR factorization of matrix X. Accurate block-size is different for each numerous problem. In this paper, we have explored a new schemes for adaptive accurate block-size under one execution of the QR factorization.

1. はじめに

固有値問題や最小二乗問題に使われる QR 分解は，線形計算の分野において最も重要な計算手法の 1 つである．この QR 分解を行う手法の 1 つに Gram-Schmidt (GS) 法がある．

†1 慶應義塾大学理工学研究科基礎理工学専攻
School of Fundamental, Graduate School of Science and Technology,Keio University
†2 慶應義塾大学理工学部
Department of Mathematics, Faculty of Science and Technology, Keio University

これまで多くの研究者が，GS 法についての研究[1),4),8),10)–12)] と，その応用についての研究[2),3),7),9),13)] を行ってきた．GS 法は，x を n 次元ベクトルとし，Q を与えられた直交行列 Q とすると，次式で表すことができる．

$$y = \left(I - QQ^T\right) x = x - QQ^T x \equiv x - Qr \tag{1}$$

式 (1) によって，行列 X の各列を順次直交化すると，次式が成立する．

$$X = QR \tag{2}$$

ただし，R は上三角行列である．GS 法の拡張の 1 つに，Block Gram-Schmidt 法 (BGS) がある．Stewart[8)] と松尾ら[11)] は，BGS 法を用いることで，GS 法と比べ，高速に QR 分解できることを示した．これは，block 化により 1 回の四則演算にかかる時間が短くなるためである．さらに，Stewart[8)] は行列 X が悪条件である場合，列の値をランダムな値に置き換える手法を提案している．また，Matsuo ら[10)] は，演算量の増加量を予測して，block-size を自動的に決定する手法を提案した．

BGS 法は，行列 X を列方向に m 分割し，分割してできた行列 X_{block} ごとに直交化する手法である．BGS の直交化は，直交行列 Q に対して次のようになる．

$$R_{12} = Q^T X_{\mathrm{block}} \tag{3}$$
$$\widehat{Y} = X_{\mathrm{block}} - QR_{12} \tag{4}$$

しかし，式 (3) と式 (4) では行列 $\widehat{Y}$ の列ベクトルどうしは直交していない．今，$\widehat{Y}, Y$ の $k\,(0 \le k \le m-1)$ 列目までを $\widehat{Y}_{1:k}, Y_{1:k}$ とし，$k+1$ 列目の列ベクトルを $\widehat{y}_{k+1}, y_{k+1}$ とする．ここで，GS 法を用いると，次式が成立する．

$$r = Y_{1:k}^T \widehat{y}_{k+1}, \quad y_{k+1} = Y_{1:k} r$$

以下これを繰り返すと，次のようになる．

$$\widehat{Y} = YR_{22} \tag{5}$$

ただし，R_{22} は上三角行列である．ここで，式 (4) と式 (5) をまとめると，X_{block} は次式を満たす．

$$X_{\mathrm{block}} = QR_{12} + YR_{22} \tag{6}$$

以上をまとめた，BGS 法のアルゴリズムを図 1 に示す．

近年，GS 法の並列化について様々な研究が行われてきた[1),2),4),5)]．これらは効果的な並列化の仕方により，高速に QR 分解できることを示した．しかし，BGS 法と同様に，block-size の問題がある．一般に，Parallel Block Gram-Schmidt 法 (PBGS) の block-size は任意に

Vol.2012-HPC-134 No.2
2012/6/1

$K := P/m$;
for $k = 0 : K-1$
  $X_{\text{block}} := X[:; km : (k+1)m]$;
  $R_{12} := Q^T X_{\text{block}}$;
  $\widehat{Y} = X_{\text{block}} - QR_{12}$;
  for l=1:m
    $r := Y_{1:l-1}^T \widehat{y}_l$;
    $y = Y_{1:(l-1)} r$;
    $R_{22}[1 : l-1; l] = r$;
    $R_{22}[k : k] = |y|$;
  end for
  $Q[:; km : (k+1)m] = Y$;
  $R = R_{12} + R_{22}$;
end for

図 1 Block Gram-Schmidt Algorithm

$K := P/m$;
for $k = 0 : K-1$
  if (myrank == 0)
    Broadcast (X_{block});
  else
    Receive (X_{block})
  $R_{12} := Q^T X_{\text{block}}$;
  $\widehat{Y} = X_{\text{block}} - QR_{12}$;
  if (myrank == 0)
    Receive ($\widehat{Y}$) from each PE;
  else
    send ($\widehat{Y}$);
end for

図 2 Parallel Block Gram-Schmidt Algorithm

決定することになる．問題によって最適な値は異なり，各問題でおのおのの block-size を決定する必要があった．最適な block-size を見つけるためには，複数回プログラムを実行し決定する必要がある．大規模な問題に対して，PBGS 法を適用する場合にはこれは非効率である．

本稿では，まず 2 章で BGS 法の並列化について述べる．そして，3 章で Block-size を適切に決定する手法を提案する．最後に，4 章と 5 章で，数値実験によって，提案手法の有効性を示す．

2. Parallel Block Gram-Schmidt 法

Vanderstraeten[1] と Gudula ら[4] は，BGS 法の並列化の手法について述べてきた．本稿では，BGS 法を Column-Wise Distribution (CWD) を使って，BGS 法を並列化した[5]．CWD は，直交化されるべき行列 X_{block} と直行行列 Q を，各 PE に記憶させる並列化の手法である．この手法は，Row-Wise Distribution (RWD) と比べて，無駄な記憶領域を使ってしまう分，実装が行いやすく，応用しやすい利点を持っている．図 2 に PBGS 法のアルゴリズムを示す．

3. Block-size の最適化

先にも述べたように，PBGS 法の block-size は任意に決定しており，最適な block-size は扱う問題によって異なる．そのため，最適な block-size を求めるにはいくつかの block-size で PBGS 法を行い，最も良い m の値を決定する必要がある．もし，規模の小さい問題ならばこのことは大きな問題ではない．しかし，1 回の計算をするだけでもとても時間がかかるような問題においては，この手法は現実的ではない．3 章では，計算時間の変化に注目し，適切な block-size を決定できる新しい手法，PBGS-m 法を提案する．

3.1 block-size と計算量

Matsuo ら[10] は，BGS 法の列ごとの QR 分解の計算時間から，全体の計算時間を推測する手法について述べた．まず，図 3 からわかるように，計算時間は線形に増加する．また，このことは計算量からも確認できる．掛け算の計算 1 回を 1 ユニットとする．$Q : n \times h$ と仮定し，$k+1$ ステップ目の PBGS 法の計算量を考える．ただし，$h = km$ とする．1 ステップあたりの PBGS 法の計算量は以下となる．

$$4nmh + 2nm(m-1) + (k+1)m^3 \tag{7}$$

これは，h に関して一次関数的に増加する． 次に，block-size の変化による計算時間の変化

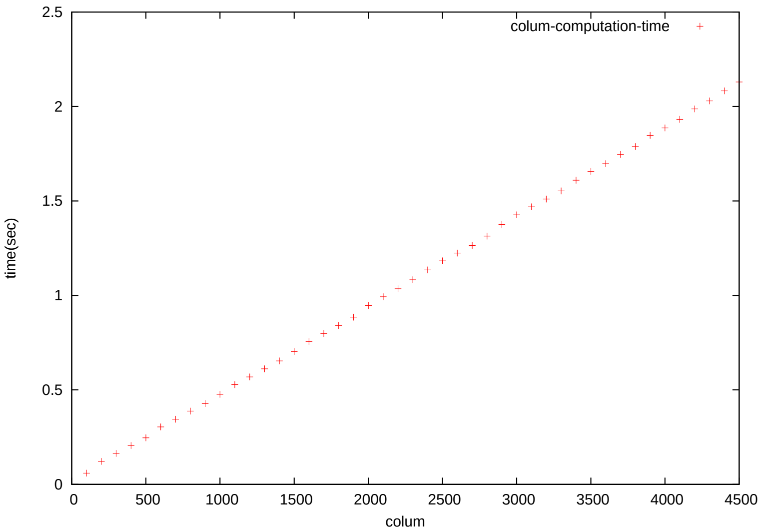

図 3 直交化と計算時間の関係

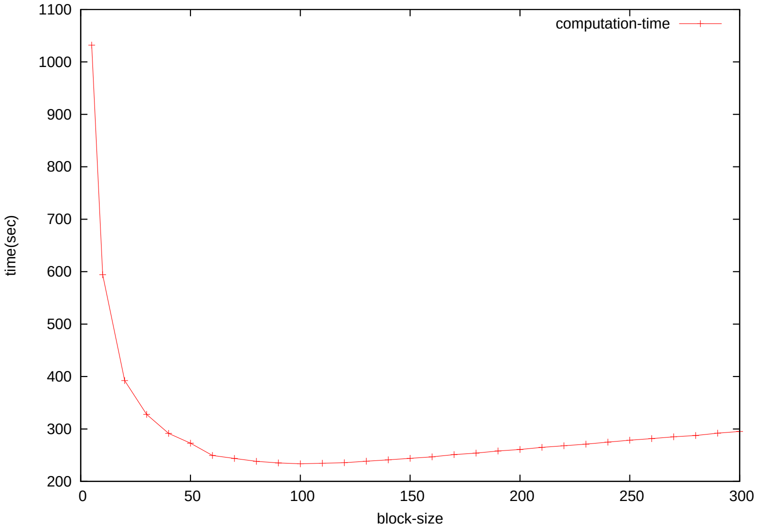

図 4 m の変化による計算時間の変化

表 1 BCSSTK18 の QR 分解

	m	time (sec)	efficiency
PBGS	10	1886	0.508
PBGS	20	1250	0.768
PBGS	50	881	1.07
PBGS-m	55	959	1.00

を示す．図 4 は，cavity-flow から生じる偏微分方程式を離散化して得られる，4562×4562 の実対称行列[6] を，block-size$m = 50$ で BGS 法により QR 分解したときの時間を，100 列ごとに計測した結果である．BGS 法は，block-size によって計算時間が大きく変わることがわかる．

3.2 適切な block-size の決定

Matsuo ら[10] に示されている提案手法の 1 つを PBGS 法に拡張し，PBGS-m 法を提案する．PBGS-m 法のアルゴリズムを図 5 に示す． まず，ステップ (01) でサンプルとして $m = 2^i, i = 1 \cdots 4$ を設定する．次に，ステップ (07)-(12) で PBGS 法でそれぞれ 1 ステップ実行し，計算時間を計測する．最後に，ステップ (13)-(18) で最適な block-size を決定する．

4. 数 値 実 験

PBGS-m 法の有効性を示すために，数値実験を行った．4 つの *Intel Xeon* 3.16GHz プロセッサを有する *SUN fire X2250* を利用し，プログラムは倍精度を用いて C 言語で記述した．扱った行列は，固有値問題から生じる実非対称疎行列 BCSSTK18 である[6]．行列サイズは 11948×11948 である．PBGS 法のいくつかの block-size による QR 分解の時間と，PBGS-m 法を用いた QR 分解の時間を比較した．実験結果を表 1 に示す．ここで，表中の efficiency は，PBGS-m 法の速度に対する割合を示す． 表 1 より，$m = 10, 20$ の場合には PBGS-m 法のほうが，高速に QR 分解できていることがわかる．特に $m = 10$ に対しては，約 2 倍近い速さである．また，$m = 50$ のときは，PBGS 法のほうが高速に計算できているがこれは PBGS-m は，自動的に block-size を決定できていることを考慮すれば，

5. ま と め

4 章より，提案手法である RBGS-m 法が有効であることが示せた．今後さらに数値実験を行い，PBGS-m 法の有効性を示すことが課題である．

Vol.2012-HPC-134 No.2
2012/6/1

Input:$X \in \mathbb{R}^{n \times n}$

Output:m

$m = 2^i, i = 1 \cdots 4$;

If (rank==0) ;

  bloadcast (X_{block});

else

  receive (X_{block});

endif

for $j = 0 : 1$;

  start:=gettimeofday();

  BGS(X_{block}) with m_i;

  end:=gettimeofday();

  $t_{ij} := end - start$;

end for

  $a = (t_{i0} - t_{i1})/m_i$

  $r[i] := \frac{1}{2} n^2 a + t_{i0} - a(h - m)$;

  $A[ij] = m_i^{j-1}$;

solve $Ax = r$

$f(m) := x_1 m^4 + x_2 m^3$
  $+ x_3 m^2 + x_4 m + x_5$;

solve optimal-m
  $:= \min_{m \in [0, \frac{1}{2} N]} f(m)$;

図 5 The PBGS-m algorithm

.

参 考 文 献

1) Vanderstraeten, D., "*An Accurae Parallel Block Gram-Schmidt Algorithm without Re-orthogonalization,*" Numer. Lin. Alg. Appl., Vol.7, pp.219–236, 2000.

2) ———, "*A Stable and Efficient Parallel Block Gram-Schmidt Algorithm,*" Euro-Par '99, LNCS 1685, pp.1128–1135, 1999.

3) Elden, L. and Park, H., "*Block Downdating of Least Squares Solutions,*" SIAM J. Matrix Anal. Appl., Vol. 15, pp. 1018–1034, 1994.

4) Gudula, R. and Michael, S., "*Comparison of Different Parallel Modified Gram-Schmidt Algorithm,*" Euro-Par 2005, LNCS 3648, pp.826–836, 2005.

5) Katagiri, T., "*Performance Evaluation of Parallel Gram-Schmidt Re-orthogonalization Methods*" VECPAR 2002, LNCS 2565, pp. 302–314, 2003.

6) "*MATRIX MARKET,*" http://math.nist.gov/MatrixMarket/, Information Technology Laboratory of the National Institute of Standards and Technology, USA.

7) Qiaohua, L., "*Modified Gram-Schmidt-based Methods for Downdating the Cholesky Factorization*" J. Comput. Appl. Math., Vol.235, pp.1897–1905, 2011.

8) Stewart, G. W., "*Block Gram-Schmidt Orthogonalization,*" SIAM J. SCI. COMPUT., Vol.31, pp. 761–775, 2008.

9) ———, "*The Effect of Rounding Errors on an Algorithm for Downdating a Cholesky Factorization*" J. Inst. Math. Appl., Vol. 23, pp. 203–213, 1979.

10) Matsuo, Y., Nodera, T., "*The Optimal Block-Size for the Block Gram-Schmidt Orthogonalization,*" J. Sci. Tech, Vol. 49, pp. 348–354, 2011.

11) ———, "ブロックグラムシュミット法を用いた *QR* 分解の高速化" 情報処理学会 第 73 回全国大会, 講演番号 4J-5, 2011.

12) ———, "*Block Gram-Schmidt* 法の適切な *block-size* の決定法" 情報処理学会 第 74 回全国大会, 講演番号 1K-1, 2012.

13) Yoo, K. and Park, H., "*Accurate Downdating of a Modified Gram-Schmidt OR Decomposition*" BIT, Vol.36, pp.166–181, 1996.