

Block Gram-Schmidt 法の適切な block-size の決定法

松尾 洋一¹, 野寺 隆²

¹ 慶應義塾大学理工学研究科, ² 慶應義塾大学理工学部
sundai7221@a5.keio.jp¹, nodera@math.keio.ac.jp²

概要

Block Gram-Schmidt 法に対する適切な block-size の決定手法を提案し, 数値実験により有用性を示す.

1 はじめに

Gram-Schmidt 法は, 正規直交基底系を構成する算法である. 今, サイズ $n \times n$ の行列 X に対し各列を順次直交化すると, 次式ようになる.

$$X = QR \quad (1)$$

ただし, R は上三角行列である.

近年, Block Gram-Schmidt 法に関連する様々な研究が行われており, その算法は表 1 のようになる. Stewart [1] や松尾ら [5] は, 行列を列ごとに分割する block 化を用いることにより, 1 回のかげ算の演算にかかる時間が短くなることを示した. さらに, block-size により QR 分解にかかる計算時間は異なるが, Stewart [1] は m を経験的に決定する算法を提案した. 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める必要があるが, これは非効率である. これを改善するために, Matsuo ら [4] 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. また Silva [2] は, block Gram-Schmidt 法に関して, 修正 Gram-Schmidt 法と Householder 変換との演算量の比較を行っている. Vanderstraeten [3] は block Gram-Schmidt 法の並列化を考案し, その算法の有効性を示した.

本発表では QR 分解を効率的に行うために最適な block-size を求める新しい手法を提案する.

2 Block Gram-Schmidt 法

2.1 計算時間

Block Gram-Schmidt 法の計算時間の変化は規則的で, 計算時間は線形に増加する (Matsuo [4]).

表 1: Block Gram-Schmidt 法

- | |
|---|
| 01 . $R_{12} = Q^T X_{\text{block}};$ |
| 02 . $\hat{Y} = X_{\text{block}} - QR_{12};$ |
| 03 . orthogonalize \hat{Y} . ($\hat{Y} = YR_{22}$); |
| 04 . $Q = (Q \ Y);$ |
| 05 . combine R_{12} and R_{22} into R . |

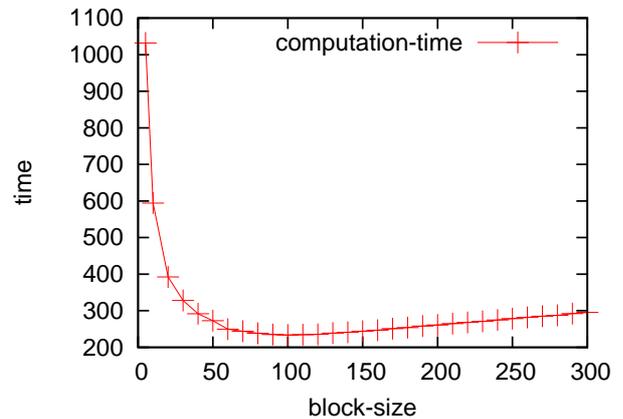


図 1: m の変化による計算時間の変化

実際, block Gram-Schmidt 法の演算量を考えると, その事実が確かめられる. ここで, 1 回の掛け算を 1 ユニットとして考える. 行列 $X: n \times n$ とし, block-size を m (簡単のために n を割り切る数とする) とする. また全ての列に対し再直交化するものと仮定する. すると, h 列目における block Gram-Schmidt 法の 1 ステップあたりの計算量は

$$4nmh + 2nm(m - 1) + (h + m)m^2 \quad (2)$$

となる. ここで, 式 (2) は, h の 1 次関数である.

次に, 図 1 に block-size の変化による計算時間の変化を示す. この図より, block-size の変化による計算時間の変化は偶関数になっていることがわかる.

2.2 適切な block-size の決定法

Block-size の変化による計算時間の変化を予測することで、プログラムの実行中に最適な block-size を自動的に決定することが可能である。

Matsuo ら [4] は、演算量から計算時間を予測する手法を提案した。しかし、 m における計算時間を 1 ステップのみから測定していたため、正確な増加量がわからなかった。また、演算量の変化のみから全体の計算時間を予測していたため、適切なものとは大きくずれた block-size を決定することがしばしばあった。

以上の事柄を改善するために、次のように適切な block-size を決定する手法を提案する。

[算法 A]

Step1 最初に、5 つの block-size $m_i, i = 1, 2, \dots, 5$ を用意する。

Step2 m_i で block- Gram-Schmidt 法を 2 ステップ実行し計算時間の増加量 t_i を計る。

Step3 $m = m_i$ の全体の計算時間 T_i を t_i から予測する。

Step4 block-size の変化による計算時間の変化を、5 つの T_i から 4 次関数で近似する。

step5 4 次関数の最小値をとる block-size を適切な block-size とする。

ここで、 m_i を小さい値にすれば、Step2 における 2 ステップの計算時間は短く、増加量も正確にわかる。よって問題が大規模な場合、計算時間全体に対する予測に必要な時間はとても小さい。また、Step3 は計算時間が線型に増加することから T_i が予測できる。Step4 は、計算時間の変化が偶関数になることを用いている。

この手法を用いることで、QR 分解の実行時間に影響を与えることなく、適切な block-size を見つけることができる。

3 数値実験

メインメモリーが 4G バイトの Sun fire X2250 を用いて、C 言語で数値実験を行った。計算精度として倍精度を用いた。固有値問題から生じる実非対称な 4 つの疎行列 [6] を QR 分解する際の、最適な block-size の大きさを比較した。結果を表 2 に示した。表中の記号 m は最適な block-size を表し、opt- m は提案手法により計算された block-size を表す。ただし、 m の値は複数回プログラムを実行

表 2: 適切な block-size と提案手法の比較

| 問題 | matrix size | m | opt-m | ratio |
|----------|---------------|-----|-------|-------|
| bcsstk02 | 112 × 112 | 90 | 51 | 0.57 |
| bcsstk06 | 420 × 420 | 40 | 26 | 0.65 |
| bcsstk15 | 3948 × 3948 | 80 | 55 | 0.69 |
| bcsstk18 | 11948 × 11948 | 100 | 41 | 0.41 |

して求めた結果である。ratio は m に対する opt- m の割合を表す。

4 結論

表 2 より、提案手法は実際の最適な block-size の 60% ~ 70% の値を取っている。これは提案手法が適切なサイズを決定できている。また、行列の大きさによって、opt- m 値は変化している。これは問題によって変わる m の値に、提案手法が対応できていることを示している。この結果、適切な block-size を計算量と計算時間から予測することは、可能である。

今後、opt- m の値を 80% 程度にすることが課題である。

参考文献

- [1] Stewart, G. W., "Block Gram-Schmidt Orthogonalization," SIAM J. Sci. Comput., Vol.31, pp. 761-775, 2008.
- [2] Silva, J. D., "Numerics of Gram-Schmidt Orthogonalization," Lin. Alg. and Its Appl. Vol. 197, 198, pp. 297-316, 1994.
- [3] Vanderstraeten, D., "An Accurate Parallel Block Gram-Schmidt Algorithm without Reorthogonalization," Numer. Lin. Alg. Appl., Vol. 7, pp. 219-236, 2000.
- [4] Matsuo, Y., and Nodera, T., "The Optimal Block-Size for the Block Gram-Schmidt Orthogonalization," J. Sci. Tech, Vol. 49, pp. 348-354, 2011.
- [5] 松尾, 野寺, 「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, 2010.
- [6] "MATRIX MARKET," <http://math.nist.gov/MatrixMarket/>, Information Technology Laboratory of the National Institute of Standards and Technology, USA.