

2P14 演題	「大規模なベクトルの記憶転送効率の良い直交化法について」	
発表者 (所属)	村上 弘 (東京都立短期大学)	
連絡先	〒196-8540 東京都昭島市東町 3 - 6 - 33 東京都立短期大学経営情報学科 TEL:042 - 543-3001(代表) FAX:042-543-3002 (共用) e-mail <murakami@tmca.ac.jp>	
キーワード	直交化、階層記憶、メモリバンド幅、参照局所性、並列化、分散記憶	
開発意図 適用分野 期待効果 特徴など	超長大なベクトル多数本の規格直交化はブロック化算法などで重要性を増している。通常の算法 MGS, HQR に比べて記憶参照の局所性が高く、記憶転送量が少ない方法を提案し考察する。誤差解析的には劣った方法だが、方法の精度上の欠点は演算精度の二倍化で容易に回避できる。記憶転送量が性能の隘路の場合には有用である。また容易に独立性の高い部分に分割でき、高度な並列化が可能である。	
環 境	適応機種名	階層記憶システム、並列分散記憶システム
	OS名	
	ソース言語	Fortran, C 等
	周辺機器	
流通形態 (右のいずれかに をつけて ください)	<ul style="list-style-type: none"> <li>・日本コンピュータ化学会の無償利用ソフトとする</li> <li>・独自に頒布する</li> <li>・ソフトハウス、出版社等から市販</li> <li>・ソフトの頒布は行なわない</li> <li>・その他                      ・ 未定</li> </ul>	具体的方法
		方法の原理と計算法を本予稿で紹介した。システムに合わせた実装は比較的簡単。

ベクトルの正規直交化は線形計算に於いて基本的である。算法を記憶の転送量を下げる為にブロック化を行う場合に正規直交化の計算が必要となる (Block Lanczos 法等)。以下に於いて、ベクトルの長さを  $N$ 、ベクトルの本数を  $m$  とし、 $m \ll N$  とする。(例えば  $m$  が  $10 \sim 1000$ 、 $N$  が  $10^3 \sim 10^9$  など。)

直交化を行うのに通常最も良く使われるものは、安定で in-place で計算できる「改良 Gram-Schmidt (MGS) 法」、あるいは計算で求められたベクトルの直交性がほぼ最良な「Householder - QR 法」である (ベクトルの長さに基づくピボット選択をすれば、尚良い)。MGS 法により得られるベクトルの正規直交からのずれは、 $\text{cond} \cdot \mu$  程度である。ここで  $\text{cond}$  はベクトルの組の条件数で、 $\mu$  はマシンイプシロンである。H-QR 法によれば得られるベクトルの直交性のずれは  $\mu$  程度になる。

通常の修正 Gram-Schmidt 法を記憶転送量に注目して考察してみると、容易に判るように第一番目のベクトル成分を除く為に  $m$  本のベクトル全体を走査し、第二番目の成分を除く為に  $m - 1$  本のベクトルを...、第  $m-1$  番目の成分を除く為に 2 本のベクトルを、と合計で約  $(1/2)m^2$  回ベクトルを走査する。あるいは配列要素の個数では約  $(1/2)Nm^2$  語であり、平均的に  $(m, N)$  型の配列を  $m/2$  回走査することに

なる。(m,N)型の配列の為の記憶容量が極めて大だとすれば、下位の階層記憶に格納せざるを得ない。上位の階層記憶(例えばCPU内部のキャッシュや主記憶)に(m,N)型配列全体が収まらず、例えばベクトル一本すらも収まらない場合、(m,N)型の配列を約m/2回ほど下位の記憶階層にアクセス(読んで書き戻すの組で1回と数えて)をおこなう必要がある。ところが記憶転送を最重視する以下の算法では、(m,N)型の配列を一回滑らかにアクセスして読み出し、その後で滑らかにアクセスして一回書き戻せば、正規直交化が可能である。但し、余分の記憶を $O(m^2)$ 程度必要とする。

**算法**(m,N)型行列Aに対し、内積 $S(I,J)$ ;  $S := AA'$  を計算する。対称行列Sの下半分は

$S(I,J) := 0$ ; for k := 1 to N do for I := 1 to m do for J := 1 to I do  $S(I,J) := S(I,J) + A(I,k) * A(J,k)$   
によりAを列方向に一回走査すれば計算できる。

m次の対称行列Sの固有分解 $S = U' U$ を求める。正值対角行列は大きい要素が先に来るよう並べる。特異値分解と同様に、地数値誤差が入るのでカットオフ値以下の要素は0とみなして得られるの一般逆行列を $\dagger$ とし、 $X := (\dagger)^{1/2} U A$ とすれば、 $XX' = I\dagger$ であるから、Xは正規直交ベクトルである。ここで $I\dagger$ は0と置いた要素に合わせて単位行列の要素を0に置いたものである。Aを列方向に一回だけ滑らかに走査しつつAの場所にXを重ね書きできることが容易に判る。Aの条件が良好ならCholesky分解 $S = LL'$ から $X = L^{-1}A$ として求める方が計算量が少ないが、(更にピボット操作を加えても)精度が劣る。

古典的行列乗算法では、AからSを求める計算量は約 $(1/2)Nm^2$ 回の乗算と加算、Sの固有分解の計算は $O(m^3)$ 、AからXを求める計算量は約 $Nm^2$ の乗算と加算である。行列高速乗算法を用いる場合、AやXの長さNの列をmずつに区切ってm次の正方行列 $N/m$ 個と余りの部分とみなし、m次の正方行列乗算が $O(m^{3-})$ で計算できるならば、AからSを更にAからXを求める演算量はどちらも $O(Nm^{2-})$ になる。古典的算法を用いる場合も行列乗算は局所性が高いので効率を稼ぐことが出来る。

**並列化** 本方法を並列化する場合には、行列Aを長さNの列方向をP個に区切り、P個の演算ユニット(PE)に作業を割り当てればよく、AからS、AからXを計算する部分は、小容量のSの部分集をまとめる箇所とUとの放送以外は完全に独立な並列計算が可能で、PE間の相互通信は起きない。

**考察** 本方法の弱点は、行列Sの条件数が行列Aの条件数の自乗になるので、MGSに比べて直交精度の低下に対するAの条件数の影響が大きく、MGSと同じ精度を常に保証するには、Sの精度とSを累積する箇所の演算演算を(通常精度のまま記憶されたAに対し)倍長で行い、Sを固有分解してU、を求めるとも倍長での演算が必要になる。AにU、を乗じる部分は通常精度の演算で行える。Aをm/2回走査する代わりに二回の走査で済むので、記憶転送が性能の隘路となる場合には、演算精度の二倍化による演算速度の数倍~十倍程度の低下は、許容しうる。

**改良反復法** 倍の精度の演算を回避し、通常精度の演算により概ね良好な解を得ることは、以下のような反復改良法で可能である。Aから $S = AA'$ を作り、Sの固有分解 $S = U' U$ によりU、を作る。Aを重ね書きしつつ $Y = UA$ を作る。Yを作る際に同時に $T = YY'$ を計算できる。これはほぼ対角行列に近い対称行列になる。間Jacobi法などでTの固有分解 $T = V V'$ を得る。そうして $X = (\dagger)^{1/2} V Y$ をYに(Aの位置に)重ね書きして作る。この全過程でAの記憶場所が列方向に三回滑らかに走査される。

**詳細文献** "A Block Orthogonalization Procedure with Constant Synchronization Requirements", Andreas Stathopoulos and Kesheng Wu, SIAM J.Sci.Comput.Vol.23,No.6,pp.2165-2182.