数理解析研究所講究録
第 1791 巻 2012 年 57-65

ハウスホルダー変換に基づく直交化法の最近の進展
～ 並列計算・高性能計算の観点から ～

山本有作
神戸大学大学院システム情報学研究科計算科学専攻／ JST-CREST

1 はじめに

任意の $m \times n$ 行列 A ($m \geq n$) は，$m \times m$ 直交行列 Q と $m \times n$ 上三角行列 R を用いて $A = QR$ と分解できる．また，Q の最初の n 列からなる行列を $\tilde{Q}$，R の最初の n 行からなる行列を $\tilde{R}$ とするとき，$A = \tilde{Q}\tilde{R}$ という分解も可能である．本稿では前者を完全 QR 分解（full QR decomposition），後者を部分 QR 分解（thin QR decomposition）と呼ぶことにする．A の列ベクトルが独立な場合，部分 QR 分解において $\tilde{R}$ の対角成分が正であるという条件を付けると，$\tilde{Q}$，$\tilde{R}$ は一意に定まる．一方，完全 QR 分解においては，R は一意に定まるが，Q は一意には定まらず．後半の $m-n$ 本の列間で直交変換を行う自由度がある．

QR 分解は，行列 $\tilde{Q}$ に着目すると，A の列ベクトルを正規直交化する手続きと見なせる．一方，行列 R に着目すると，直交行列により A を上三角化する手続きと見なせる．どちらの目的のためにも，QR 分解は数値計算において頻繁に利用される．QR 分解を行うには多くの手法があるが，本発表では，数値的安定性に優れたハウスホルダー変換による手法を取り上げる．

QR 分解は，用途に応じて様々な条件での利用が考えられる．具体的には，

(i) 直交行列 Q（または $\tilde{Q}$）を陽的に求める必要があるか，

(ii) 直交行列 Q（あるいは Q^T）を，後で他の行列に作用させる必要があるか，

(iii) A の列ベクトルが最初から全部与えられるか，あるいは直交化の過程で 1 本ずつ与えられるか，

などの条件が考えられる．本発表では，実際の数値計算アルゴリズムでよく出てくる組合せとして，次のような条件下での QR 分解を考える．

(A) A は最初から全部与えられ，$\tilde{Q}$ を陽的に求める．Q を後で他の行列に作用させる必要はない．

(B) A は最初から全部与えられる．Q，$\tilde{Q}$ を陽的に求める必要はない．Q^T を後で他の行列に作用させる．

(C) A の列ベクトルは直交化の過程で 1 本ずつ与えられ，$\tilde{Q}$ を陽的に求める．Q を後で他の行列に作用させる必要はない．

(A) は標準的なベクトルの正規直交化である．たとえば第一原理分子動力学法では，各時間ステップで波動関数を直交化するため，このタイプの直交化を行う．また，ブロック版のクリロフ部分空間法による連立 1 次方程式解法でも，このタイプの直交化が用いられる．(B) は，たとえばブロック行列 $C = [A, B]$ ($B \in \mathbf{R}^{m \times l}$) において，$A$ を QR 分解した後，$Q^T B$ を計算する問題に相当する．この計算は，ブロック化された行列の QR 分解や，固有値計算のための帯行列化などで現れる．(C) は，GMRES 法など，アーノルディ過程に基づく行列計算で典型的に現れる．アーノルディ過程では，反復ごとにクリロフ部分空間が拡張されて新しい基底ベクトルが生成され，それを今までに求めた正規直交基底に対して直交化する．これがちょうど (C) のタイプの直交化となる．このタイプの直交化は，固有ベクトル計算のための逆反復法や MR^3 法において，固有値が密集する場合にも現れる．

近年では，問題の大規模化に伴い，並列計算機や GPU（グラフィックスプロセッサ）など，高性能計算技術の利用が重要となっている．従来，QR 分解あるいは直交化は，並列化が難しい演算と見なされ，高性能計算におけるボトルネックと見なされることが多かった．しかし最近では，新しいアルゴリズムの開発などにより，様々なタイプの QR 分解において，効率的な計算が可能となっている．そこで本発表では，上記 (A)，(B)，(C) の 3 つのタイプの QR 分解に対し，高性能計算の観点から，最近のトピックスを紹介する．

2 タイプ (A)： TSQR 法とその後退誤差解析

タイプ (A) の計算は，ハウスホルダーに基づく標準的な QR 分解アルゴリズムにより実行できる．ただし，このアルゴリズムを並列化すると，ハウスホルダー変換ごとに 2 回のプロセッサ間同期（あるいはプロセッサ間通信）が必要となり，全体で $2n$ 回の同期が必要となって，オーバーヘッドが大きい．これに対して Langou は，行列 A を上下に 2 つの部分行列 $A_1, A_2 \in \mathbf{R}^{(m/2)\times n}$ に分割してそれぞれの QR 分解を計算し，それらの結果を合成して A の QR 分解を求める方法を提案した [1]. これを TSQR（Tall and Skinny QR）アルゴリズムまたは AllReduce アルゴリズムと呼ぶ．アルゴリズムは次のように書ける．

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad A_1 = Q_1R_1, \quad A_2 = Q_2R_2, \tag{1}$$

$$\bar{A} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}, \quad \bar{A} = \bar{Q}\bar{R}. \tag{2}$$

ここで，式 (1) の第 2 式，第 3 式，式 (2) の第 2 式はそれぞれ A_1，A_2，$\bar{A}$ のハウスホルダー QR 分解であり，$Q_i \in \mathbf{R}^{(m/2)\times(m/2)}$，$R_i \in \mathbf{R}^{(m/2)\times n}$ $(i = 1, 2)$，$\bar{Q} \in \mathbf{R}^{m\times m}$，$\bar{R} \in \mathbf{R}^{m\times n}$ である．式 (1)，(2) より，A は次のように書ける．

$$A = \begin{bmatrix} Q_1 & O \\ O & Q_2 \end{bmatrix} \bar{Q}\bar{R}. \tag{3}$$

右辺の最初の 2 つの行列は直交行列であるから，これは A の QR 分解を与える．

$\bar{A}$ は，第 1 行から第 n 行，第 $m/2+1$ 行から第 $m/2+n$ 行がそれぞれ上三角行列，それ以外の要素がすべてゼロの行列である．この非ゼロ構造を考慮すると，$\bar{A}$ の QR 分解は実質的に $(2n)\times n$ 行列の QR 分解となる．したがって，$m \gg n$ の場合，この部分の演算量は小さく，演算量の大部分は A_1，A_2 の QR 分解で占められる．これらの QR 分解は並列に計算できるため，2 個のプロセッサによる大粒度並列化が可能となる．また，A_1，A_2 の QR 分解に対しても TSQR 法を再帰的に適用することで，さらに多くのプロセッサを計算に利用できる．

以上のように，TSQR 法は並列計算に極めて適した手法であり，多くの応用が期待されている [2]. しかし，TSQR 法では，プロセッサ数が 2^L 個のとき，ハウスホルダー変換を L 回再帰的に行う必要があり，丸め誤差が増大しないかが心配になる．これに対して我々は，TSQR 法の後退誤差解析を行い，TSQR 法を有限精度で実行して得られる QR 分解の因子 $\hat{Q}$，$\hat{R}$ が次の式を満たすことを示した [4].

$$A + \Delta A = Q\hat{R}, \quad \|\Delta A\|_F \simeq n\gamma_{c\cdot\frac{m}{2^L}} + Ln\gamma_{c\cdot 2n}, \tag{4}$$

$$\hat{Q} = Q + \Delta Q, \quad \|\Delta Q\|_F \simeq \left(n\gamma_{c\cdot\frac{m}{2^L}} + Ln\gamma_{c\cdot 2n}\right)\sqrt{n}. \tag{5}$$

ここで，Q はある厳密に直交な行列（定義は [4] 参照）であり，ΔA，ΔQ は誤差行列である．また，

$$\gamma_{cm} = \frac{cm\mathbf{u}}{1 - cm\mathbf{u}} \tag{6}$$

であり，$\mathbf{u}$ はマシンイプシロン，c はある正の定数である．式 (4)，(5) はそれぞれ，TSQR 法が後退誤差の意味で安定であること，計算された $\hat{Q}$ が直交行列に極めて近いことを示す．なお，これらの式で $L = 0$ とすると，よく知られたハウスホルダー QR 分解の後退誤差解析の結果 [3] が得られる．これらの式で注目すべきは，L があまり大きくない範囲では，誤差限界 $\|\Delta A\|_F$，$\|\Delta Q\|_F$ が L に関して減少関数となっていることである．すなわち，TSQR 法は，L が比較的小さい範囲では，従来のハウスホルダー QR 分解よりむしろ高精度となりうる．この結果は数値実験でも確認されており，TSQR 法の応用を拡げるための基盤になると考えられる．

3 タイプ (B)： TSQR 法で生成されるコンパクト WY 表現の合成

本節では，TSQR 法で計算した Q を使って，$m \times l$ 行列 B に対し，Q^TB を計算する問題を扱う．計算機環境として，最近普及してきたマルチコアプロセッサと GPU のハイブリッド環境を用いることを考える．

この場合，A の QR 分解は比較的複雑な計算であるから，マルチコア上で行い，Q^TB の計算は，単純な計算であるから，GPU 上で行うのが自然である．しかし，前者に TSQR 法を用いた場合，得られる Q は，TSQR 法の再帰構造に対応して，$L+1$ 個の直交行列を掛け合わせた形になり（$L=1$ の場合が式 (3)），さらに，各直交行列は多数の細かい非ゼロブロックを持つ行列となる．このような行列に対する Q^TB の計算では，ベクトル長が短くなり，GPU などのアクセラレータの性能を引き出しにくい．

そこで，Q^T を 1 つの大きな行列として表現し直すことを考える [5]．従来のハウスホルダー QR 分解の場合，適当な $m\times n$ 行列 Y，$n\times n$ 下三角行列 T を用いて $Q^T = I_m - YTY^T$（I_m は m 次単位行列）と表現できることが知られており [6]，これをコンパクト WY 表現と呼ぶ．コンパクト WY 表現を作成しておけば，Q^TB の計算は，ハウスホルダー変換を 1 個ずつ作用させるのと同じ $4mnl$ の計算量で，しかも行列乗算のみを使って計算できるので，高性能計算の観点から有利である．そこで，TSQR 法の場合でも，適当な $m\times n$ 行列 Y，$n\times n$ 行列 S を用いて，$I_m - YSY^T$ の形の行列を構成することを考える．

ただし，TSQR 法で得られる Q を直接この形に表現しようとすると，Y の列数は一般に n では収まらない．これは，この行列が実質的に（すなわち恒等変換としてでなく）作用する部分空間の次元が，通常のハウスホルダー変換の場合と異なり，n より大きくなるからである．そこで，制限を緩め，A を $\bar{R}$ に変換する直交行列で，適当な $Y\in\mathbf{R}^{m\times n}$，$S\in\mathbf{R}^{n\times n}$ を用いて $I_m - YSY^T$ の形に書ける行列を求めることにする．これは，完全 QR 分解における Q の非一意性を利用していることに相当する．

いま，Q の最初の n 列からなる行列を $\tilde{Q} = [\tilde{Q}_1^T\ \tilde{Q}_2^T]^T$（$\tilde{Q}_1\in\mathbf{R}^{n\times n}$，$\tilde{Q}_2\in\mathbf{R}^{(m-n)\times n}$）とし，

$$Y = \begin{bmatrix} \tilde{Q}_1 - I_n \\ \tilde{Q}_2 \end{bmatrix}, \quad S = (I_n - \tilde{Q}_1)^{-1} \tag{7}$$

とおく．すると，この Y と S は条件を満たす [5]．実際，

$$\begin{aligned} (I_m - YSY^T)A &= \bar{R}, &(8)\\ (I_m - YSY^T)^T(I_m - YSY^T) &= I_m &(9) \end{aligned}$$

が成り立つことが容易にわかる．こうして，求める表現が構成できた．この書き換えは，$\tilde{Q}$ の計算のために約 $2mn^2$ の余分な演算量を必要とするものの，Q^TB の計算を大型の行列乗算に置き換えられることから，ハイブリッド環境での計算には有効となる可能性がある．

4 タイプ (C)：コンパクト WY 表現に基づく逐次添加型の直交化

ベクトル $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \mathbf{R}^m$（$m\geq n$）に対し，それらを正規直交化したベクトル $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ を求める問題を考える．ただし，ベクトル $\mathbf{a}_i$（$2\leq i\leq n$）は最初から与えられるのではなく，$\mathbf{q}_1, \ldots, \mathbf{q}_{i-1}$ がわかって初めて計算できるとする．これを，ベクトル逐次添加型の直交化と呼ぶ．

ベクトル逐次添加型の直交化では，修正グラム・シュミット法（MGS 法）がよく使われる．しかし，MGS 法は計算に逐次性があることに加え，行列 $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ の条件数を $\kappa(A)$ とするとき，$\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ の直交性が $\kappa(A)$ に比例して悪化するという問題点がある．一方，古典的グラム・シュミット法（CGS 法）を 2 回繰り返す直交化法も提案されている．この手法は，並列性および $\{\mathbf{q}_i\}$ の直交性に関しては優れているが，条件 $O(\mathbf{u}\kappa(A)) < 1$ が成り立たないと適用できないという問題点がある [2]．

これに対して，Walker [7] は，GMRES 法においてグラム・シュミット法の代わりにハウスホルダー変換に基づく直交化を用いるアルゴリズムを提案している．この手法は，$\kappa(A)$ によらずに高い直交性を確保できる点に特徴がある．さらに [7] では，複数のハウスホルダー変換を合成することで，直交化演算を行列ベクトル積の形に変形し，並列粒度を高める方法も提案されている．しかし，後者については，直交性などの数値的安定性に関する理論的検討は行われておらず，また，並列計算機上での性能評価も行われていない．

我々は，Walker の後者の方法をハウスホルダー変換のコンパクト WY 表現を用いて書き直し，これが通常のハウスホルダー変換と同等の数値的安定性を持つことを示した [8]．また，この方法が GMRES 法に限らず一般のベクトル逐次添加型の直交化に利用できることを指摘するとともに，並列計算機上で実際に高い加速率を達成できることを示した．そのため，この方法は，クリロフ部分空間法による行列指数関数 $\exp(C)\mathbf{x}$ の計算，固有値が密集する場合の逆反復法，MR^3 法など，様々なアルゴリズムにおいて活用できると考えられる．以下，この方法について説明し，数値的安定性の解析を行う．

4.1 ハウスホルダー変換に基づくベクトル逐次添加型の直交化

準備として，ハウスホルダー変換に基づくベクトル逐次添加型の直交化 [9] を Algorithm 1 に示す．ここで，$\mathrm{House}_i(\mathbf{x})$ は，ベクトル $\mathbf{x}$ に対し，その最初の $i-1$ 個の要素を不変にし，第 $i+1$ 要素以下を 0 にするハウスホルダー変換 $P_i = I - t_i \mathbf{y}_i \mathbf{y}_i^T$（$I$ は単位行列）を求める操作である．また，$\mathbf{e}_i$ は単位行列の第 i 列ベクトルを表す．

```
[Algorithm 1]
do i = 1, n
   Generate a_i from q_1, q_2, ..., q_{i-1}.
   a'_i = P_{i-1} ··· P_2 P_1 a_i
   H_i = House_i(a'_i)
   q_i = P_1 P_2 ··· P_i e_i
end do
```

ここで $\mathbf{q}_i$ の計算では，QR 分解の定義より $\mathbf{q}_i$ が行列 $P_1P_2\cdots P_n$ の第 i 列であること，および $P_{i+1},\ldots,P_n$ による乗算が $\mathbf{e}_i$ を不変にすることを用いた．このアルゴリズムでは，$\mathbf{a}'_i$ および $\mathbf{q}_i$ の生成において，複数のハウスホルダー変換を順次適用していくため，MGS 法と同じ逐次性がある．

4.2 コンパクト WY 表現

一方，複数のハウスホルダー変換 $P_k = I - t_k \mathbf{y}_k \mathbf{y}_k^T$（$1 \le k \le i$）の作用をまとめて行う方法として，コンパクト WY 表現 [6] がある．コンパクト WY 表現では，$Y_1 = [\mathbf{y}_1]$，$T_1 = [t_1]$ として，$n \times k$ 行列 Y_k，$k \times k$ 下三角行列 T_k を次のように帰納的に定義する．

$$Y_k = [Y_{k-1}\ \mathbf{y}_k], \tag{10}$$

$$T_k = \begin{bmatrix} T_{k-1} & \mathbf{0} \\ -t_k \mathbf{y}_k^T Y_{k-1} T_{k-1} & t_k \end{bmatrix}. \tag{11}$$

すると，$P_i \cdots P_2 P_1$ の積は次のように表現できる．

$$P_i \cdots P_2 P_1 = I - Y_i T_i Y_i^T. \tag{12}$$

これにより，複数のハウスホルダー変換を順次適用する計算を，行列ベクトル積として実行できる．

4.3 コンパクト WY 表現に基づくベクトル逐次添加型の直交化

コンパクト WY 表現を使うと，Algorithm 1 を行列ベクトル積を用いて書き直すことができる．アルゴリズムは次のようになる．

[Algorithm 2]
do $i = 1, n$
  Generate $\mathbf{a}_i$ from $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{i-1}$.
  $\mathbf{a}'_i = (I - Y_{i-1} T_{i-1} Y_{i-1}^T)\mathbf{a}_i$
  $(t_i, \mathbf{y}_i) = \mathrm{House}_i(\mathbf{a}'_i)$
  $Y_i = [Y_{i-1}\ \mathbf{y}_i]$
  $T_i = \begin{bmatrix} T_{i-1} & \mathbf{0} \\ -t_i \mathbf{y}_i^T Y_{i-1} T_{i-1} & t_i \end{bmatrix}$
  $\mathbf{q}_i = (I - Y_i T_i^T Y_i^T)\mathbf{e}_i$
end do

Algorithm 2 は，[7] における直交化部分の計算を GMRES 法に特化しない形に切り出し，コンパクト WY 表現を用いて書き直したものに相当する．$\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{i-1}$ から $\mathbf{a}_i$ を計算する方法は任意であり，ベクトル逐次添加型の直交化を用いる任意のアルゴリズムに適用できる．主要な演算はすべて行列ベクトル積の形なので，MGS 法や Algorithm 1 に比べて粒度の大きい並列化が可能となる．

4.4 数値的安定性

本項では，Algorithm 2 について数値的安定性の解析を行う．以下，ベクトル $\mathbf{x}$ に対し，その各要素の絶対値を要素とするベクトルを $|\mathbf{x}|$ で表す．また，$|\mathbf{x}| \le |\mathbf{y}|$ のような不等式は，要素ごとの不等式を表すとする．まず準備として，ハウスホルダー変換および WY 表現に対する既存の誤差解析の結果を述べる．なお，WY 表現に対する結果は [3] で与えられているが，証明や誤差限界における定数の詳細が省かれているため，ここでは証明も含めて詳しく述べる．

補題 1（[3]，Lemma 18.3） $\mathbf{v}_i$（$1 \le i \le r$）を正確なハウスホルダー変換のベクトルとし，条件

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \Delta\mathbf{v}_i, \quad |\Delta\mathbf{v}_i| \le \gamma_{cm}|\mathbf{v}_i| \tag{13}$$

を満たす $\mathbf{v}_i$ の近似ベクトルを $\hat{\mathbf{v}}_i$ とする．また，$P_i = I - \mathbf{v}\mathbf{v}^T$ とし，$U_i = P_i \cdots P_2 P_1$ とする．いま，$\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_r$ を用いた（近似的な）ハウスホルダー変換を浮動小数点演算により行列 $A \in \mathbf{R}^{m\times n}$ に作用させて得られる結果を $\hat{A}_{r+1}$ とする．このとき，

$$\hat{A}_{r+1} = U_r(A + \Delta A), \quad \|\Delta A\|_F \le r\gamma_{cm}\|A\|_F \tag{14}$$

が成り立つ．

補題 2（[3], Section 18.4） $\mathbf{v}_i$，$\hat{\mathbf{v}}_i$，P_i，U_i は補題 1 と同様とする．このとき，$\{\hat{\mathbf{v}}_k\}_{k=1}^{i}$ を用いて浮動小数点演算で求めた WY 表現を $\hat{U}_i = I + \hat{W}_i\hat{Y}_i^T$ とすると，$i = 1, 2, \ldots, r$ について次の式が成り立つ．

$$\hat{U}_i = I + \hat{W}_i\hat{Y}_i^T = P_i \cdots P_2 P_1 + \Delta\hat{U}_i = U_i + \Delta U_i, \quad \|\Delta U_i\|_F \le d_1(m, i)\mathbf{u}. \tag{15}$$

ただし，$d_1(m, i)$ は m，i に関して低次の多項式オーダーで増大する関数である．

証明　まず $i = 1$ のときを考えると，

$$\begin{aligned} \hat{U}_1 &= I - \hat{\mathbf{v}}\hat{\mathbf{v}}^T = I - (\mathbf{v}_i + \Delta\mathbf{v}_i)(\mathbf{v}_i + \Delta\mathbf{v}_i)^T \\ &= U_1 + \mathbf{v}_i\Delta\mathbf{v}_i^T + \Delta\mathbf{v}_i\mathbf{v}_i^T + \Delta\mathbf{v}_i\Delta\mathbf{v}_i^T \\ &\equiv U_1 + \Delta U_1. \end{aligned} \tag{16}$$

よって，

$$\begin{aligned} \|\Delta U_1\|_F &\le \|\mathbf{v}_1\Delta\mathbf{v}_1^T\|_F + \|\Delta\mathbf{v}_1\mathbf{v}_1^T\|_F + \|\Delta\mathbf{v}_1\Delta\mathbf{v}_1^T\|_F \\ &= 2\|\mathbf{v}_1\|_2\|\Delta\mathbf{v}_1\|_2 + \|\Delta\mathbf{v}_1\|_2^2 \\ &\simeq 2\gamma_{cm} \end{aligned} \tag{17}$$

次に，ΔU_{i-1} が与えられたときに ΔU_i を求める漸化式を導く．まず，WY 表現の定義より，次の式が成り立つ．

$$\begin{aligned} \hat{U}_i &= I + \hat{W}_i\hat{Y}_i^T = I + \begin{bmatrix} \hat{W}_{i-1} & -\hat{\mathbf{v}}_i \end{bmatrix} \begin{bmatrix} \hat{Y}_{i-1}^T \\ fl(\hat{\mathbf{v}}_i^T\hat{U}_{i-1}) \end{bmatrix} \\ &= I + \hat{W}_{i-1}\hat{Y}_{i-1}^T - \hat{\mathbf{v}}_i fl(\hat{\mathbf{v}}_i^T\hat{U}_{i-1}) \\ &= \hat{U}_{i-1} - \hat{\mathbf{v}}_i fl(\hat{\mathbf{v}}_i^T\hat{U}_{i-1}). \end{aligned} \tag{18}$$

この最後の式は，行列 $\hat{U}_{i-1}$ に $I - \hat{\mathbf{v}}\hat{\mathbf{v}}^T$ を浮動小数点演算により作用させた場合の結果の式

$$fl(\hat{U}_{i-1} - fl(\hat{\mathbf{v}}_i fl(\hat{\mathbf{v}}_i^T\hat{U}_{i-1}))) \tag{19}$$

に似ているが，むしろそれより丸め誤差が入る機会が少ない．したがって，補題1の誤差評価がそのまま使える．補題1において $r=1$，$A=\hat{U}_{i-1}$ とすると，最後の式は次のように書ける．

$$\hat{U}_{i-1}-\hat{\mathbf{v}}_i fl(\hat{\mathbf{v}}_i^T\hat{U}_{i-1})=P_i(\hat{U}_{i-1}+\Delta\hat{U}_{i-1}),\quad \|\Delta\hat{U}_{i-1}\|_F\le\gamma_{cm}\|\hat{U}_{i-1}\|_F. \tag{20}$$

これを式 (18) に代入し，式 (15) で $i-1$ とした式を代入すると，

$$\begin{aligned}\hat{U}_i &= P_i(\hat{U}_{i-1}+\Delta\hat{U}_{i-1})\\ &= P_i(U_{i-1}+\Delta U_{i-1}+\Delta\hat{U}_{i-1})\\ &= U_i+P_i(\Delta U_{i-1}+\Delta\hat{U}_{i-1})\\ &\equiv U_i+\Delta U_i.\end{aligned} \tag{21}$$

これより，ΔU_i に関する漸化式が次のように得られる．

$$\Delta U_i=P_i(\Delta U_{i-1}+\Delta\hat{U}_{i-1}). \tag{22}$$

これを解くと，次の式が得られる．

$$\Delta U_i=\sum_{j=1}^{i-1}P_iP_{i-1}\cdots P_{j+1}\Delta\hat{U}_j+P_iP_{i-1}\cdots P_2\Delta U_1. \tag{23}$$

ただし，$P_iP_{i-1}\cdots P_{j+1}$ のような行列の積は，最初の添字より最後の添字のほうが大きい場合は，単位行列を表すとする．これより，

$$\begin{aligned}\|\Delta U_i\|_F &\le \sum_{j=1}^{i-1}\|P_iP_{i-1}\cdots P_{j+1}\Delta\hat{U}_j\|_F+\|P_iP_{i-1}\cdots P_2\Delta U_1\|_F\\ &= \sum_{j=1}^{i-1}\|\Delta\hat{U}_j\|_F+\|\Delta U_1\|_F\\ &\le \gamma_{cm}\sum_{j=1}^{i-1}\|\hat{U}_j\|_F+\|\Delta U_1\|_F\\ &\le \gamma_{cm}\sum_{j=1}^{i-1}\|U_j\|_F+\gamma_{cm}\sum_{j=1}^{i-1}\|\Delta U_j\|_F+\|\Delta U_1\|_F\\ &\simeq \{(i-1)\sqrt{m}+2\}\gamma_{cm}+\gamma_{cm}\sum_{j=1}^{i-1}\|\Delta U_j\|_F.\end{aligned} \tag{24}$$

ここで，漸化式 $u_i=\{(i-1)\sqrt{m}+2\}\gamma_{cm}+\gamma_{cm}\sum_{j=1}^{i-1}u_j$ の解は次のように与えられる．

$$\begin{aligned}u_i &= \sum_{j=1}^{i}(\gamma_{cm})^{i-j+1}\{(i-1)\sqrt{m}+2\}\\ &= \gamma_{cm}\left\{\sqrt{m}\cdot\frac{\gamma_{cm}^i-i\gamma_{cm}+i-1}{(1-\gamma_{cm})^2}+2\cdot\frac{1-\gamma_{cm}^i}{1-\gamma_{cm}}\right\}.\end{aligned} \tag{25}$$

これは m,i に関して低次の多項式オーダーの関数である．一方，$\|\Delta U_i\|_F$ は，この漸化式において等号を不等号に変えた式を満たすから，その値は式 (25) の最右辺以下に抑えられる．(証明終)

注意1 本補題では，$\hat{U}_i=I+\hat{W}_i\hat{Y}_i^T$ と定義している．すなわち，$\hat{W}_i$，$\hat{Y}_i$ それぞれの誤差は考えるが，それらを掛け合わせたり，I に加えたりするときに生じる丸め誤差は考えない．WY 表現は，$I+\hat{W}_i\hat{Y}_i^T$ という行列を陽的に計算してから，それを作用させるわけではないからである．WY 表現を作用させる際の誤差については，補題4で考察する．

注意 2 本補題の証明で鍵となる式は (18) である．この式は [3] で与えられている．しかし，厳密に考えるとこの式には疑問がある．なぜなら，この式では行列 $\hat{U}_{i-1}$ を実在の行列として扱い，それに $\hat{\mathbf{v}}_i$ を用いたハウスホルダー変換を作用させると考えて補題 1 を適用しているが，実際の計算はこのようには行わないからである．WY 表現の生成における実際の計算では，$\hat{\mathbf{v}}_i^T\hat{U}_{i-1}$ を求めるのに，$\hat{U}_{i-1}$ を陽的には計算せず，

$$fl(\hat{\mathbf{v}}_i^T + fl(fl(\hat{\mathbf{v}}_i^T\hat{W}_{i-1})\hat{Y}_{i-1}^T)) \tag{26}$$

のように計算を行う．このような計算に対して補題 1 をそのまま適用できるかどうかは，検討を要する．ただし本稿では，式 (18) が正しいと考えて誤差解析を行う．

注意 3 ここでは WY 表現に対する補題を示したが，同様の補題がコンパクト WY 表現についても成り立つ．以下の補題についても同様である．

補題 2 より，直ちに Algorithm 2 で計算したベクトル $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ の直交性に関する次の命題が導ける．

系 3 $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ を Algorithm 2 で計算したベクトルとする．このとき，$m \times n$ 列直交行列 $\tilde{Q}$ および $m \times n$ 行列 $\Delta\tilde{Q}$ が存在して，

$$[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n] = \tilde{Q} + \Delta\tilde{Q}, \quad \|\Delta\tilde{Q}\|_F \le d_1(m,n)\mathbf{u} \tag{27}$$

が成り立つ．すなわち，$\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ の正規直交系からのずれは，$A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ の条件数によらず，$O(\mathbf{u})$ である．

証明 Algorithm 2 での $\mathbf{q}_i$ の計算方法より，$\mathbf{q}_i$ は $I - Y_nT_n^TY_n^T$ の第 i 列となる（Y_i，T_i の非ゼロ構造より，$(I - Y_iT_i^TY_i^T)\mathbf{e}_i = (I - Y_nT_n^TY_n^T)\mathbf{e}_i$ に注意）．したがって $[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]$ は $I - Y_nT_n^TY_n^T$ の部分行列となるから，補題 2 と注意 3 より，明らかに主張が成り立つ．(証明終)

補題 2 は WY 表現の直交性に関する補題であったが，WY 表現の後退安定性については次の補題が成り立つ．

補題 4 $B \in \mathbf{R}^{m\times l}$ とし，浮動小数点演算により $I - \hat{W}_i\hat{Y}_i^T$ を B に作用させて得られる行列を C とする．このとき，行列 $\Delta C \in \mathbf{R}^{m\times l}$ が存在して，次の式が成り立つ．

$$C = U_iB + \Delta C = U_i(B + U_i^T\Delta C), \tag{28}$$

$$\|\Delta C\|_F \le [1 + d_1(i,m) + d_2(i,m)d_3(i,m)(1 + c_1(i,m,l) + c_1(m,i,l))]\mathbf{u}\|B\|_F + O(\mathbf{u}^2). \tag{29}$$

ここで，U_i は補題 1 で定義した行列であり，d_1, d_2, d_3 は i と m のみに依存する定数，c_1 は i，n，l のみに依存する定数である．

補題 4 は，補題 2，行列乗算に対する誤差解析結果 [3]，および $\|\hat{W}_i\|_F$，$\|\hat{Y}_i\|_F$ に対する上界の評価から導かれる．$\hat{W}_i$ は長さ $\sqrt{2}$ のベクトル $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_i$ を並べた行列であるから，$\|\hat{W}_i\|_F = O(\sqrt{n})$ は明らかである．一方，$\hat{Y}_i$ は式 (18) により定義されるため，この式を用いてノルムを評価できる．なお，[3] では 2 ノルムに対して同様の式を導いているが，ここでは前の補題との関係から，フロベニウスノルムで統一した．

補題 4 を用いると，Algorithm 2 の後退誤差に関する次の定理が導ける．

定理 5 $R \in \mathbf{R}^{m\times n}$ を，第 (i,j) 要素（$i \le j$）が $(I - t_j\mathbf{y}_j\mathbf{y}_j^T)\mathbf{a}'_j$ の第 i 要素で，それ以外の要素がゼロの行列とする．すなわち，R は浮動小数点演算を用いて Algorithm 2 により計算した $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ の R 因子とする．このとき，行列 $\Delta A \in \mathbf{R}^{m\times n}$ が存在して次の式が成り立つ．

$$A + \Delta A = U_n^TR, \quad \|\Delta A\|_F \le d_4(m,n)\mathbf{u}\|A\|_F. \tag{30}$$

ここで，U_n は補題 1 で定義した行列であり，$d_4(m,n)$ は m と n のみに依存する定数である．

証明　まず，アルゴリズムより $\mathbf{a}'_j = (I - Y_{j-1}T_{j-1}Y_{j-1}^T)\mathbf{a}_j$ である．そこで，補題 4 (をコンパクト WY 表現用に直した補題) より，あるベクトル $\Delta\mathbf{a}_j$ と定数 $d'_4(m,j)$ が存在して，

$$\mathbf{a}'_j = U_{j-1}(\mathbf{a}_j + \Delta\mathbf{a}_j), \quad \|\Delta\mathbf{a}_j\|_2 \leq d'_4(m,j)\mathbf{u}\|\mathbf{a}_j\|_2 \tag{31}$$

が成り立つ．さらに，R の第 j 列を $\mathbf{r}_j \in \mathbf{R}^m$ とすると，補題 1 より，あるベクトル $\Delta\mathbf{a}'_j$ が存在して，

$$\mathbf{r}_j = P_j(\mathbf{a}'_j + \Delta\mathbf{a}'_j), \quad \|\Delta\mathbf{a}'_j\|_2 \leq \gamma_{cm}\|\mathbf{a}'_j\|_2 \simeq \gamma_{cm}\|\mathbf{a}_j\|_2 \tag{32}$$

が成り立つ．そこで，これらの式より，

$$\begin{aligned} \mathbf{r}_j &= P_j(U_{j-1}(\mathbf{a}_j + \Delta\mathbf{a}_j) + \Delta\mathbf{a}'_j) \\ &= U_j(\mathbf{a}_j + (\Delta\mathbf{a}_j + U_{j-1}^T\Delta\mathbf{a}'_j)) \\ &\equiv U_j(\mathbf{a}_j + \Delta\mathbf{a}''_j), \qquad (33) \\ \|\Delta\mathbf{a}''_j\|_2 &\leq (d'_4(m,j)\mathbf{u} + \gamma_{cm})\|\mathbf{a}_j\|_2 \\ &\equiv d''_4(m,j)\mathbf{u}\|\mathbf{a}_j\|_2. \qquad (34) \end{aligned}$$

さらに，$\mathbf{r}_j$ は第 $j+1$ 要素以降がゼロであるから，$P_nP_{n-1}\cdots P_{j+1}$ をかけても変化しないことを用いると，

$$\mathbf{r}_j = U_n(\mathbf{a}_j + \Delta\mathbf{a}''_j). \tag{35}$$

これらを $j = 1, 2, \ldots, n$ についてまとめて書くと，

$$R = U_n(A + \Delta A), \quad \Delta A = [\Delta\mathbf{a}''_1, \Delta\mathbf{a}''_2, \ldots, \Delta\mathbf{a}''_n]. \tag{36}$$

ここで，

$$\|\Delta A\|_F \leq \left\{\max_{1\leq j\leq n} d''_4(m,j)\right\}\mathbf{u}\sqrt{\sum_{j=1}^{n}\|\Delta\mathbf{a}''_j\|_2^2} = \left\{\max_{1\leq j\leq n} d''_4(m,j)\right\}\mathbf{u}\|\Delta A\|_F. \tag{37}$$

したがって，

$$d_4(m,n) = \max_{1\leq j\leq n} d''_4(m,j) \tag{38}$$

とおくと，定理が得られる．(証明終)

定理 5 は，Algorithm 2 が後退安定であることを意味する．

4.5　他の手法との比較

コンパクト WY 表現に基づくベクトル逐次添加型の直交化手法を他の手法と比較した結果を表に示す．

	MGS	CGS2	House	cWY
計算量	$2n^2m$	$4n^2m$	$4n^2m$	$4n^2m$
同期回数	$O(n^2)$	$O(n)$	$O(n^2)$	$O(n)$
直交性	$O(\mathbf{u}\kappa(A))$	$O(\mathbf{u})$	$O(\mathbf{u})$	$O(\mathbf{u})$
条件	–	$O(\mathbf{u}\kappa(A)) < 1$	–	–

ここで，CGS2 は CGS 法を 2 回繰り返す方法，House はハウスホルダー変換に基づく Algorithm 1 の方法，cWY はコンパクト WY 表現を用いた Algorithm 2 の方法を表す．また，直交性は，$Q = [\mathbf{q}_1, \ldots, \mathbf{q}_n]$ としたときの行列 $Q^TQ - I_n$ のノルムを表す．条件とは，この直交性を実現するために必要な，行列 $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ に関する条件である．MGS，CGS2，House に関する結果は [2] によった．

表より，コンパクト WY 表現に基づく手法は，演算量は修正グラム・シュミット法に比べて多いものの，CGS2，Algorithm 1 とは同等であり，並列化のための同期回数が少なく，かつ，任意の $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ に対して適用できるという長所を持つ手法と位置付けられる．

5 おわりに

ハウスホルダー変換に基づくQR分解は，精度と安定性に優れた手法であるが，高性能計算の観点からは，これまでボトルネックと見なされることが多かった．しかし，最近ではアルゴリズムの進展により，様々なタイプのQR分解について効率的な計算ができるようになっている．本発表では3つのタイプのQR分解を取り上げ，このことを示した．今後の課題としては，様々な行列計算アルゴリズムへの組み込みと性能・精度評価，各アルゴリズムに対するより完全な理論誤差解析，計算機環境・問題サイズに応じたアルゴリズムの自動チューニングなどが挙げられる．

謝辞

研究集会「科学技術計算における理論と応用の新展開」において発表の機会を下さった山本野人先生，小林健太先生に感謝いたします．また，有益なコメントを下さった同研究集会の参加者の皆様に感謝いたします．本研究は科学研究費補助金の補助を受けている．

References

[1] J. Langou: AllReduce algorithms: application to Householder QR factorization, presented at *Preconditioning 2007*, Toulouse, France, July 9-12, 2007.

[2] J Demmel, L. Grigori, M. Hoemmen and J. Langou: Communication-optimal parallel and sequential QR and LU factorizations, *LAPACK Working Notes*, No. 204, 2008.

[3] N. Higham: *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2002.

[4] D. Mori, Y. Yamamoto and S. -L. Zhang: Backward error analysis of the AllReduce algorithm for Householder QR decomposition, to appear in *Japan Journal of Industrial and Applied Mathematics.*

[5] 山本有作: TSQR アルゴリズムで生成される Compact WY 表現の合成について, 日本応用数理学会 2011 年度年会講演予稿集, pp. 95–96 (2011).

[6] R. Schreiber and C. van Loan: A storage-efficient WY representation for products of Householder transformations, *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, No. 1 pp. 53–57 (1989).

[7] H. Walker: Implementation of the GMRES method using Householder transformations, *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, No. 1, pp. 152-163 (1988).

[8] Y. Yamamoto and Y. Hirota: A parallel algorithm for incremental orthogonalization based on the compact WY representation, *JSIAM Letters*, Vol. 3, pp. 89-92 (2011).

[9] G. Golub and C. van Loan: *Matrix Computations*, Johns Hopkins Univ. Press, 1996.