

CMSI計算科学技術特論A

# 行列計算における高速アルゴリズム

- 大規模固有値問題の数値解法 -

2013年6月27日

神戸大学大学院システム情報学研究科 計算科学専攻

山本有作



# 目次

---

- 固有値問題の基礎
  - 固有値問題の分類
  - 固有値・固有ベクトルの性質
  - 固有値計算のための枠組み
- べき乗法
  - アルゴリズム
  - べき乗法からの発展
- Arnoldi法
  - アルゴリズム
  - 反復に伴う演算量・記憶領域の増加
  - リスタートとその問題点
  - Implicitly Restarted Arnoldi法



# 目次

---

- Lanczos法
  - アルゴリズム
  - 丸め誤差と再直交化
  - リスタートとその問題点
  - Thick Restart Lanczos法
- 終わりに
- 参考文献



# 固有値問題の分類

- 行列の対称性

- 実対称行列:  $A = A^T$
- 複素エルミート行列:  $A = \bar{A}^T$ 
  - $\bar{A}^T$  を  $A^*$  または  $A^H$  と書く
- 非対称 / 非エルミート行列: 上記以外

- 標準固有値問題とその一般化

- 標準固有値問題:  $Ax = \lambda x$
- 一般固有値問題:  $Ax = \lambda Bx$
- 2次固有値問題:  $\lambda^2 Ax + \lambda Bx + Cx = \mathbf{0}$
- 非線形固有値問題:  $A(\lambda)x = \mathbf{0}$

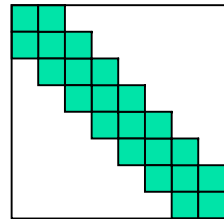
} 本講義では扱わない

# 固有値問題の分類(続き)

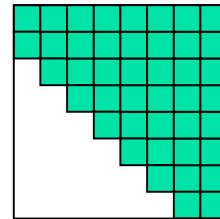
## ■ 行列の疎性

- 密行列
- 疎行列
- 特別な行列
  - 実対称3重対角行列
  - ヘッセンベルグ行列

} 固有値・固有ベクトルを効率的に  
計算可能



3重対角行列



ヘッセンベルグ行列

## ■ 求めたい固有値・固有ベクトルの数

- 全固有値・固有ベクトル
- 一部の固有値・固有ベクトル



# 実対称行列の固有値問題: $A = A^T$

- 固有値
  - 固有値は**実数**
- 固有ベクトル
  - $n$  本の固有ベクトルが**正規直交系**をなすように取れる

$$\begin{aligned} A\mathbf{u}_i &= \lambda_i\mathbf{u}_i \quad (i = 1, 2, \dots, n), \\ \mathbf{u}_i^T \mathbf{u}_j &= \delta_{ij} \quad (i, j = 1, 2, \dots, n). \end{aligned}$$

- 直交行列により対角化可能

$$U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n] \text{ とおくと,}$$

$$U^T A U = \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n).$$

- 複素エルミート行列の固有値問題:  $A = A^*$ 
  - 実対称行列とほぼ同じ性質を持つ



# 非対称行列の固有値問題

- 固有値

- 固有値は一般に複素数
- ただし,  $A$  が実行列の場合, 複素固有値は共役のペアで現れる

- 固有ベクトル

- $n$  本の1次独立な固有ベクトルが取れるならば,  $A$  は対角化可能

$A\mathbf{x}_i = \lambda_i\mathbf{x}_i \quad (i = 1, 2, \dots, n)$  において,  $\{\mathbf{x}_i\}$  が1次独立ならば,

$X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  とおくと,

$$X^{-1}AX = \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n).$$

- 特に,  $n$  個の固有値がすべて異なるならば, 対角化可能
  - 異なる固有値に属する固有ベクトルは1次独立
- 縮重固有値を持つ行列は, **対角化不可能**な場合もある



## 非対称行列の固有値問題(続き)

- $A^T$  の固有値・固有ベクトル

- $A$  と  $A^T$  とは同じ固有値を持つ
- $\lambda_i$  が  $A$  の固有値のとき, 次の式を満たすベクトル  $y_i$  が存在する

$$\begin{aligned} A^T y_i &= \lambda_i y_i \\ y_i^T A &= \lambda_i y_i^T \end{aligned}$$

- これを,  $A$  の固有値  $\lambda_i$  に属する **左固有ベクトル** と呼ぶ

- Schur 標準形

- 任意の行列  $A$  は, 直交(またはユニタリ)行列  $U$  による相似変換で上三角行列  $T$  に変換できる。これを  $A$  の **Schur 標準形** と呼ぶ

$$U^T A U = T$$

- $T$  の対角要素は  $A$  の固有値となる





# 相似変換と固有値

## ■ 相似変換

- 正則行列  $Z$  を用いて,  $C = Z^{-1}AZ$  と相似変換を行ったとする
- このとき,  $C$  の固有値は  $A$  の固有値と同じ
- 固有値  $\lambda_i$  に属する  $A$  の固有ベクトル  $y_i$  と,  $C$  の固有ベクトル  $w_i$  との間には次の関係がある

$$x_i = Zw_i$$

## ■ 固有値計算での利用

- この性質を利用し,  $A$  をまず固有値計算に適した中間形に相似変換してから, 固有値・固有ベクトルを求めることが行われる
- 中間形としては, **3重対角行列**, **ヘッセンベルグ行列**が使われる



# 一般固有値問題 $Ax = \lambda Bx$ の解法

- $A$  が対称,  $B$  が対称・正定値・良条件の場合
  - $B = LL^T$  と分解し, 標準固有値問題  $L^{-1}AL^{-T}y = \lambda y$  に帰着させる
  - 元の問題の固有値は  $\lambda$ , 固有ベクトルは  $x = L^{-T}y$
- $B$  が良条件の場合
  - 標準固有値問題  $B^{-1}Ax = \lambda x$  に帰着させる
- それ以外の場合
  - $B$  のコレスキー分解や  $B^{-1}A$  を陽的に求めず, それと等価な操作を陰的に行う解法を用いる
    - QZ法など

# 固有値の特徴付け：実対称行列の場合

## ■ 最大固有値

- 実対称行列  $A$  の固有値を  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  とすると、最大固有値  $\lambda_1$  は次のように特徴付けられる

$$\lambda_1 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^T A \mathbf{x}.$$

## ■ 証明

- $A$  の固有ベクトルを用いて  $\mathbf{x}$  を  $\mathbf{x} = \sum_{i=1}^n c_i \mathbf{v}_i$  ( $\sum_{i=1}^n c_i^2 = 1$ ) と展開すると、  
 $\mathbf{x}^T A \mathbf{x} = (\sum_{i=1}^n c_i \mathbf{v}_i)^T (\sum_{i=1}^n c_i \lambda_i \mathbf{v}_i) = \sum_{i=1}^n c_i^2 \lambda_i$ 。これは、 $c_1 = 1, c_2 = \dots = c_n = 0$  のときに最大となり、最大値は  $\lambda_1$ 。

## ■ 最小固有値

- 最小固有値  $\lambda_n$  は次のように特徴付けられる

$$\lambda_n = \min_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \min_{\|\mathbf{x}\|_2=1} \mathbf{x}^T A \mathbf{x}.$$



# 固有値の特徴付け：実対称行列の場合

- Courant-Fischer の最大・最小定理
  - 実対称行列  $A$  に対し,  $r$  番目に大きい固有値  $\lambda_r$  は次のように特徴付けられる

$$\begin{aligned}\lambda_r &= \max_{S: \dim S=r} \min_{\mathbf{x} \in S, \mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\ &= \min_{S: \dim S=n-r+1} \max_{\mathbf{x} \in S, \mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.\end{aligned}$$

ただし,  $S$  は指定された次元を持つ  $\mathbb{R}^n$  の部分空間である

- $\mathbf{x}^T A \mathbf{x} / \mathbf{x}^T \mathbf{x}$  を  $x$  の  $A$  に関する Rayleigh 商と呼ぶ



# 固有値の摂動

---

- 問題

- 行列が  $A \rightarrow A + \delta A$  と変化したときの固有値の変化  $\delta\lambda$  を評価する

- 非対称行列の場合

- $\lambda$  の右・左固有ベクトルをそれぞれ  $x, y$  とすると, 次式が成り立つ

$$\delta\lambda = \frac{\mathbf{y}^T \delta A \mathbf{x}}{\mathbf{y}^T \mathbf{x}} + O(\|\delta A\|_2^2).$$

したがって,  $x$  と  $y$  が直交に近い固有値は, **悪条件**である

- 実対称行列の場合

- 次式が成り立つ

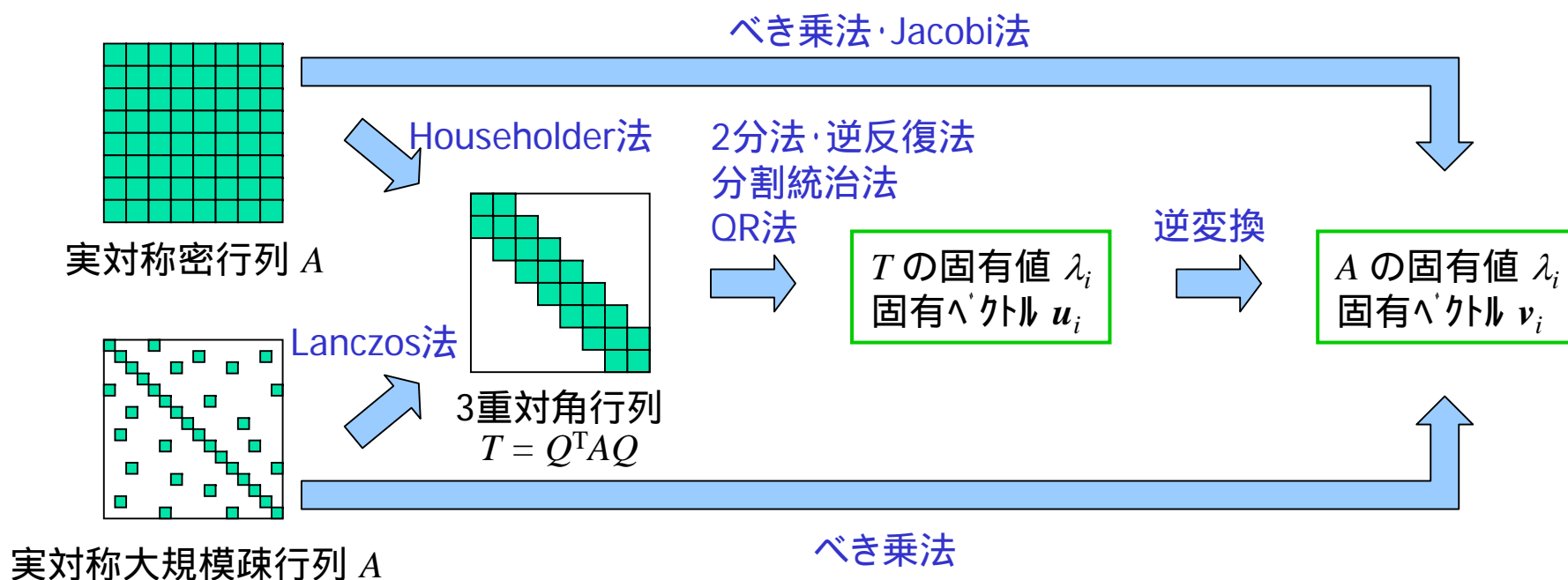
$$\delta\lambda = \|\delta A\|_2.$$

したがって, 固有値は常に **良条件**である

# 固有値計算の枠組み：実対称行列

## ■ 解法の分類

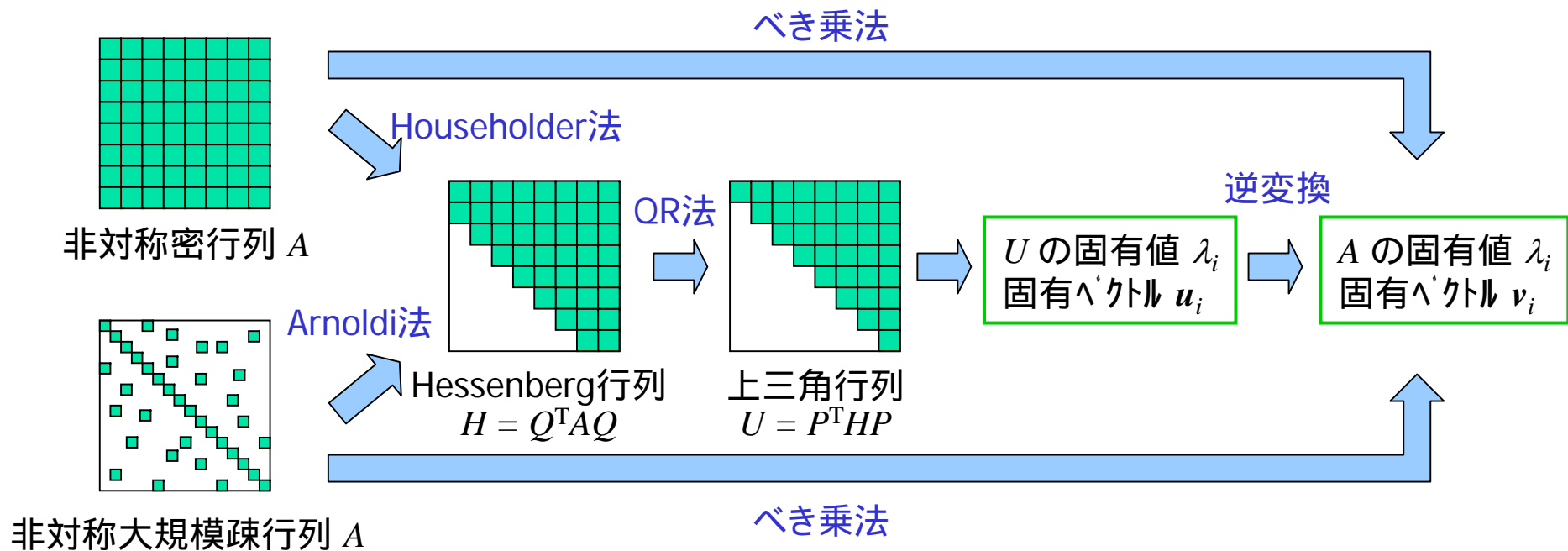
- $A$  の固有値・固有ベクトルを直接求める方法：べき乗法, Jacobi法
- 3重対角行列  $T$  を経由して固有値・固有ベクトルを求める方法
  - 正確な3重対角化：Householder法
  - 近似的な3重対角化：Lanczos法
  - 3重対角行列の固有値計算：2分法・逆反復法, 分割統治法, QR法



# 固有値計算の枠組み：非対称行列

## ■ 解法のカテゴリ

- $A$  の固有値・固有ベクトルを直接求める方法：べき乗法
- Hessenberg行列  $H$  を経由して固有値・固有ベクトルを求める方法
  - 正確なHessenberg化：Householder法
  - 近似的なHessenberg化：Arnoldi法
  - Hessenberg行列の固有値計算：QR法





# べき乗法

- アイディア

- 任意の初期ベクトル  $x^{(0)}$  から出発し, 行列  $A$  を繰り返し掛ける
- これにより, 絶対値最大の固有値に対応する固有ベクトル成分が優越し, 反復ベクトル  $x^{(k)}$  はその固有ベクトルに近づく

- アルゴリズム

- オーバーフローを避けるため, 毎回  $x^{(k)}$  を規格化する

```
[Power method]
Choose  $x^{(0)}$  arbitrarily.
for  $k = 1, 2, 3, \dots$ 
     $v = Ax^{(k-1)}$ 
     $x^{(k)} = v / \|v\|_2$ 
end for
```



# べき乗法 (続き)

## 収束証明

- 簡単のため,  $A$  が対角化可能で,  $|\lambda_1| > |\lambda_i|$  ( $i \geq 2$ ) と仮定する
- $x^{(0)}$  を  $A$  の固有ベクトルで  $x^{(0)} = \sum_{i=1}^n c_i v_i$  と展開すると,  $c_1 \neq 0$  ならば,

$$\begin{aligned} \mathbf{x}^{(k)} &= \frac{A^k \mathbf{x}^{(0)}}{\|A^k \mathbf{x}^{(0)}\|_2} \\ &= \frac{\sum_{i=1}^n c_i \lambda_i^k \mathbf{v}_i}{\sqrt{\sum_{i=1}^n \sum_{j=1}^n c_i c_j \lambda_i^k \lambda_j^k \mathbf{v}_i^T \mathbf{v}_j}} \\ &= \frac{\sum_{i=1}^n \frac{c_i}{|c_1|} \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i}{\sqrt{\sum_{i=1}^n \sum_{j=1}^n \frac{c_i}{c_1} \cdot \frac{c_j}{c_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k \left(\frac{\lambda_j}{\lambda_1}\right)^k \mathbf{v}_i^T \mathbf{v}_j}} \\ &\rightarrow \frac{c_1}{|c_1|} \mathbf{v}_1 \end{aligned}$$

- よって,  $x^{(0)}$  は  $\pm v_1$  に収束する



# べき乗法の特徴

---

- アルゴリズムの簡単さ
  - 必要な演算は、行列ベクトル積と、ベクトルの規格化のみ
  - 行列を変形しないため、記憶領域は  $A$  の非ゼロ要素数分のみ
    - 大規模疎行列の場合に有利
- 適用範囲
  - このままの形では、絶対値最大の固有値に対応する固有ベクトルしか求められない
- 収束性
  - 収束性は  $|\lambda_1|/|\lambda_2|$  ( $\lambda_2$  は絶対値が2番目に大きい固有値) に依存
  - この値が1に近いと収束が遅い



# べき乗法からの発展

- 逆反復法
  - $A$  の代わりに  $(A - sI)^{-1}$  ( $s$  はある複素数) に対してべき乗法を適用
  - $s$  に最も近い固有値に対応する固有ベクトルを計算可能
- Lanczos法/Arnoldi法
  - $x^{(0)}, x^{(1)}, \dots, x^{(k)}$  全部の情報を利用して固有ベクトルを計算
    - べき乗法では,  $x^{(k)}$  のみから固有ベクトルを計算
  - べき乗法に比べ, 収束性が向上
  - 絶対値最大の固有値以外に属する固有ベクトルも計算可能
- 直交化付き同時反復法・QR法
  - 複数の初期ベクトルに対して, 同時にべき乗法を行う
    - ただし, 同じ固有ベクトルへの収束を避けるため, 直交化を行う
  - $n$  本の初期ベクトルを用いれば, 全固有値・固有ベクトルを計算可能



# Anoldi 法：大規模非対称疎行列向け解法

## ■ アイディア

- $x^{(0)}, x^{(1)}, \dots, x^{(k)}$  全部の情報を利用して固有ベクトルを計算する
- 数値的安定性のため,  $x^{(0)}, x^{(1)}, \dots, x^{(k)}$  そのものではなく, それらを **正規直交化**したベクトル  $q^{(1)}, q^{(2)}, \dots, q^{(k)}$  を使って計算を行う
  - $q^{(1)}, q^{(2)}, \dots, q^{(k)}$  はクリロフ部分空間  $K_k(A; x^{(0)})$  の正規直交基底
  - Arnoldi 過程により生成できる
- $K_k(A; x^{(0)})$  中で固有ベクトルを定めるため, Ritz-Galerkin法を用いる
  - 次の式を満たすように近似固有値  $\lambda$ , 近似固有ベクトル  $x$  を定める

$$r = Ax - \lambda x \perp K_k(A; x^{(0)})$$

# Arnoldi 法 ( 続き )

- クリロフ部分空間の正規直交基底の生成
  - Arnoldi 過程をそのまま用いればよい

[Arnoldi process]

$$\mathbf{q}_1 = \mathbf{x}^{(0)} / \|\mathbf{x}^{(0)}\|_2$$

for  $k = 1, 2, 3, \dots$

$$\mathbf{v} = A\mathbf{q}_k$$

for  $j = 1$  to  $k$

$$h_{jk} = \mathbf{q}_j^T \mathbf{v}$$

$$\mathbf{v} = \mathbf{v} - h_{jk}\mathbf{q}_j$$

end for

$$h_{k+1,k} = \|\mathbf{v}\|_2$$

$$\mathbf{q}_{k+1} = \mathbf{v} / h_{k+1,k}$$

end for

} 新たなベクトルの生成

} 直交化

} 正規化

# Arnoldi 法 (続き)

## ■ Arnoldi 分解 (復習)

- Arnoldi 過程 より,  $Aq_i$  は  $q_1, q_2, \dots, q_{i+1}$  を使って次のように書ける

$$Aq_i = h_{1i}q_1 + h_{2i}q_2 + \dots + h_{i+1,i}q_{i+1}$$

- そこで,  $Q_k = [q_1 | q_2 | \dots | q_k]$ , ( $n \times k$  列直交行列)

$$\tilde{H}_k = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1k} \\ h_{21} & h_{22} & \dots & h_{2k} \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{kk} \\ & & & h_{k+1,k} \end{bmatrix} \quad ((k+1) \times k \text{ 行列})$$

と定義すると, 次の式が成り立つ

$$AQ_k = Q_{k+1}\tilde{H}_k$$

- $A$  の  $k$  次の **Arnoldi 分解**

# Arnoldi 法 (続き)

- Arnoldi 分解 (続き)

- $AQ_k = Q_{k+1}\tilde{H}_k$  に左から  $Q_n^T$  をかけると,

$$\boxed{Q_k^T A Q_k = H_k.} \quad (Q_k \in \mathbf{R}^{k \times n}, H_k \in \mathbf{R}^{k \times k})$$

ただし,

$$Q_k = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k], \quad K_k(A; \mathbf{x}^{(0)}) \text{ の正規直交基底}$$

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{kk} \end{bmatrix}.$$

ヘッセンベルグ行列



# Arnoldi 法 (続き)

- 近似固有値・固有ベクトルの決定

- $x \in K_k(A; x^{(0)})$  より,  $x = Q_k y$  ( $y \in \mathbf{R}^k$ ) とおく

- Ritz-Galerkin 条件より,

$$\begin{aligned} Q_k^T(Ax - \lambda x) &= Q_k^T A Q_k y - \lambda Q_k^T Q_k y \\ &= H_k y - \lambda y \\ &= \mathbf{0}. \end{aligned}$$

したがって,  $k \times k$  行列に対する固有値問題

$$\boxed{H_k y = \lambda y} \quad \text{を解けばよい}$$

これは, **ヘッセンベルグ行列**に対する固有値問題となる

- 近似固有値は  $\lambda$ , 近似固有ベクトルは  $x = Q_k y$  で与えられる





# Arnoldi 法 (続き)

## ■ アルゴリズムのまとめ

- Arnoldi 過程により  $Q_k, H_k$  を生成
- $k \times k$  行列の固有値問題  $H_k \mathbf{y} = \lambda \mathbf{y}$  を解く
  - 小規模固有値問題なので, LAPACK (QR法) などで解く
- 近似固有値として  $\lambda$ , 近似固有ベクトルとして  $\mathbf{x} = Q_k \mathbf{y}$  を採用

## ■ 演算量

- Arnoldi 過程:  $O(k^2 n)$
- $k \times k$  行列の固有値問題:  $O(k^3)$
- 近似固有ベクトルの計算:  $O(kn) \times (\text{必要な固有ベクトル本数})$
- Arnoldi 法は通常  $k \ll n$  として使うため, Arnoldi 過程の部分が演算量のほとんどを占める



# Arnoldi 法の問題点

---

- 演算量・記憶領域の増大
  - 演算量:  $O(k^2n)$
  - 記憶領域:  $O(k^2n)$
- リスタート
  - 反復回数  $k$  に上限値を設定
  - 上限値まで反復しても収束しなければ,  $q_k$  を  $q_1$  として, 再び最初から Arnoldi 法を実行
- 問題点
  - ベクトル  $q_1, q_2, \dots, q_{k-1}$  に蓄えられた情報をすべて捨ててしまう
  - 収束が遅くなる



# Implicitly Restarted Arnoldi 法 (1)

## ■ 基本的なアイデア

- リスタート時に,  $K_k(A; x^{(0)})$  中の  $l$  本 ( $l < k$ ) のベクトルを保存
- $l$  本のベクトル  $q_1^+, q_2^+, \dots, q_l^+$  は, 求めたい固有値に対応する固有ベクトルの成分が大きくなるよう,  $q_1, q_2, \dots, q_k$  の線形結合として選ぶ

$$[q_1^+, q_2^+, \dots, q_l^+] = [q_1, q_2, \dots, q_k]U \quad (U \in \mathbf{R}^{k \times l}; \text{列直交行列})$$

## ■ メリット

- リスタートを行っても,  $l$  本分のベクトルの情報が保存される
- 空間の次元の上限が  $k$  に固定      演算量・記憶領域の削減

## ■ 課題

- どんな線形結合を作ればよいか?
- Arnoldi 分解の形  $AQ_l^+ = Q_{l+1}^+ \tilde{H}_l^+$  を保つには?

## Implicitly Restarted Arnoldi 法 (2)

- $q_1^+, q_2^+, \dots, q_l^+$  の計算方法

- 入力:  $k$ -step Arnoldi 分解  $AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T$
- シフト  $\mu_1, \mu_2, \dots, \mu_{k-l}$  を用い,  $H_k$  に対して  $k-l$  ステップのQR法を実行すると, 次の新しい関係式を得る

$$AQ_k^+ = Q_k^+ H_k^+ + h_{k+1,k} q_{k+1} e_k^T U$$

ただし  $Q_k^+ = Q_k U$ ,  $H_k^+ = U^T H_k U$ ,  
 $U = U_1 U_2 \dots U_{k-l}$  (QR法の直交変換の行列)

- 両辺の第  $l$  列目までを取ると, 次の新しい  $l$ -step Arnoldi 分解が得られる

$$AQ_l^+ = Q_l^+ H_l^+ + h_{l+1,l} q_{l+1} e_l^T$$

- この状態から再度 Arnoldi 法を始めることにより,  $l$  本のベクトルを保存してリスタートすることが可能

# Implicitly Restarted Arnoldi 法 (3)

## QR法実行後の関係式

$$A \times \begin{array}{|c|} \hline Q_k^+ \\ \hline \end{array} = \begin{array}{|c|} \hline Q_k^+ \\ \hline \end{array} \times \begin{array}{|c|} \hline H_k^+ \\ \hline \end{array} + \begin{array}{|c|} \hline h_{k+1,k} \mathbf{q}_{k+1} \\ \hline \end{array} \times \begin{array}{|c|} \hline \mathbf{e}_k^T \\ \hline \end{array} \times \begin{array}{|c|} \hline U \\ \hline \end{array}$$

$\begin{array}{|c|} \hline l \quad k-l \\ \hline \end{array}$ 
 $\begin{array}{|c|} \hline k-l+1 \\ \hline \end{array}$

$$A \times \begin{array}{|c|} \hline Q_k^+ \\ \hline \end{array} = \begin{array}{|c|} \hline Q_k^+ \\ \hline \end{array} \times \begin{array}{|c|} \hline H_k^+ \\ \hline \end{array} + \begin{array}{|c|} \hline h_{k+1,k} \mathbf{q}_{k+1} \mathbf{e}_k^T U \\ \hline \end{array}$$

## 新しい $l$ -step Arnoldi 分解

$$A \times \begin{array}{|c|} \hline Q_l^+ \\ \hline \end{array} = \begin{array}{|c|} \hline Q_l^+ \\ \hline \end{array} \times \begin{array}{|c|} \hline H_l^+ \\ \hline \end{array} + \begin{array}{|c|} \hline h_{l+1,l}^+ \mathbf{q}_{l+1}^+ \mathbf{e}_l^T \\ \hline \end{array}$$

$\begin{array}{|c|} \hline k \\ \hline \end{array}$ 
 $\begin{array}{|c|} \hline l \\ \hline \end{array}$

# Implicitly Restarted Arnoldi 法 (4)

- シフトと初期ベクトルの関係

- QR法におけるシフトを  $\mu_1, \mu_2, \dots, \mu_{k-1}$  とすると,

$$UR = (H_k^+ - \mu_1 I) (H_k^+ - \mu_2 I) \dots (H_k^+ - \mu_{k-1} I)$$

ただし,  $R = R_{k-1} \dots R_2 R_1$  ( $R_i$  はQR法で現れる上三角行列)

- したがって,  $Q_k^+$  の第1列を  $q_1^+$  とすると,

$$\begin{aligned} q_1^+ &= Q_k U e_1 \\ &= Q_k (H_k^+ - \mu_1 I) (H_k^+ - \mu_2 I) \dots (H_k^+ - \mu_{k-1} I) R^{-1} e_1 \\ &= r_{11}^{-1} (A - \mu_1 I) (A - \mu_2 I) \dots (A - \mu_{k-1} I) q_1 \end{aligned}$$

- したがって, implicit restart を行うことは, 初期ベクトルを

$$q_1 \quad (A - \mu_1 I) (A - \mu_2 I) \dots (A - \mu_{k-1} I) q_1$$

と変えたことと等価



# Implicitly Restarted Arnoldi 法 (5)

---

- シフトの選び方

- $\mu$  をシフトとして選ぶと, 初期ベクトル  $q_1$  のうち,  $\mu$  に近い固有値に対応する固有ベクトル成分が小さくなる
- したがって, シフトは, **必要でない固有ベクトル**の固有値に近い値を選ぶ



# アルゴリズム

Arnoldi 分解  $AQ_k = Q_k H_k + h_{k+1,k} \mathbf{q}_{k+1} \mathbf{e}_k^T$  を入力。

DO  $l = 1, 2, 3, \dots$

シフト  $\mu_1, \mu_2, \dots, \mu_{k-l}$  を用い,  $H_k$  に対して  $k-l$  ステップのQR法を実行

$$U = U_1 U_2 \dots U_{k-l}$$

$$Q_k^+ = Q_k U, \quad H_k^+ = U^T H_k U$$

$$\beta_l = (H_k^+)_{l+1,l}, \quad \sigma_l = U_{kl}$$

$$\mathbf{q}_{l+1}^+ = \mathbf{q}_{l+1} \beta_l + h_{k+1,k} \mathbf{q}_{k+1} \sigma_l$$

$$h_{l+1,l} = \|\mathbf{q}_{l+1}^+\|_2$$

$$\mathbf{q}_{l+1}^+ := \mathbf{q}_{l+1}^+ / h_{l+1,l}$$

$Q_k^+$  の最初の第  $l$  列からなる行列を  $Q_l^+$  とする。

$H_k^+$  の左上部分からなる  $l \times l$  行列を  $H_l^+$  とする。

Arnoldi 分解  $AQ_l^+ = Q_l^+ H_l^+ + h_{l+1,l} \mathbf{q}_{l+1}^+ \mathbf{e}_l^T$  から始めて, Arnoldi法を  $k-l$  ステップ行う。

END DO





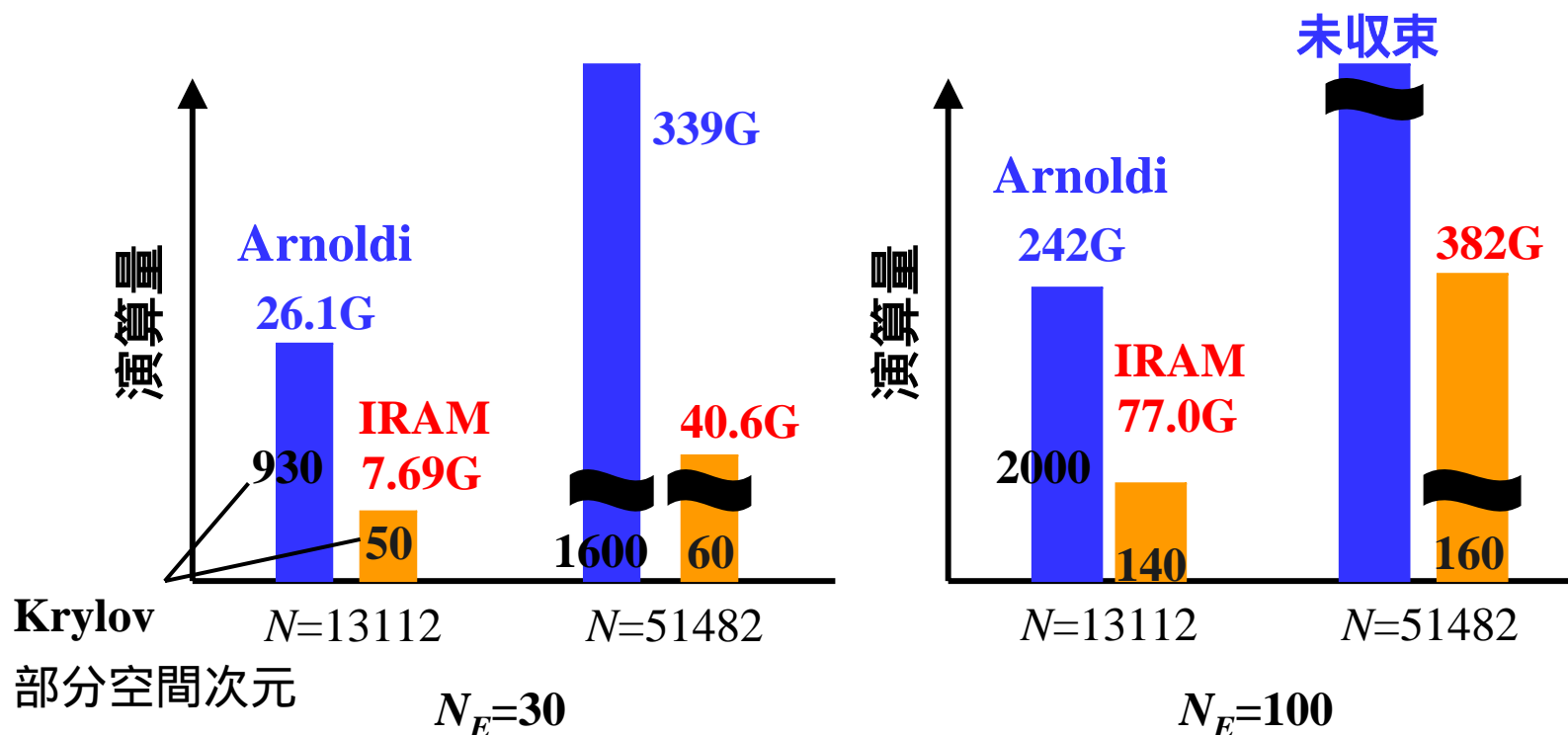
# 性能評価 (1)

---

- 評価環境
  - SR8000 (F1)
- 評価例題
  - 光導波路のベクトル波モード解析
- 評価対象の手法
  - Arnoldi法
  - IRAM (ARPACK)
- 評価指標
  - 収束に必要な演算量
  - 収束状況

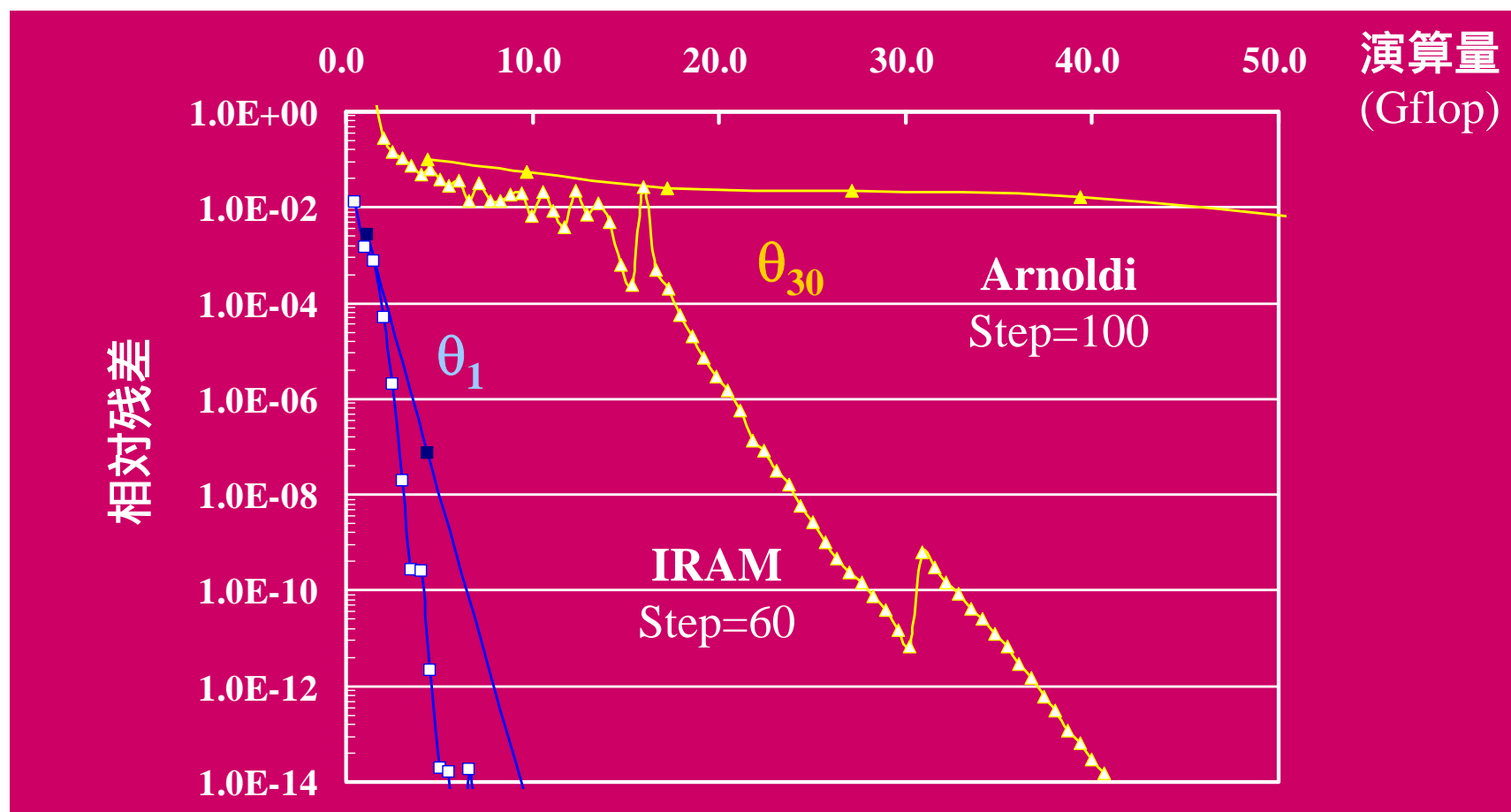
## 性能評価 (2)

- 収束判定基準  $\|Ax - x\theta\| / \|\theta\| < 10^{-14}$  (相対残差)
- 行列サイズ  $N = 13112, 51482$
- 所望固有値個数  $N_E = 30, 100$
- 結果 **IRAMは収束性優良**



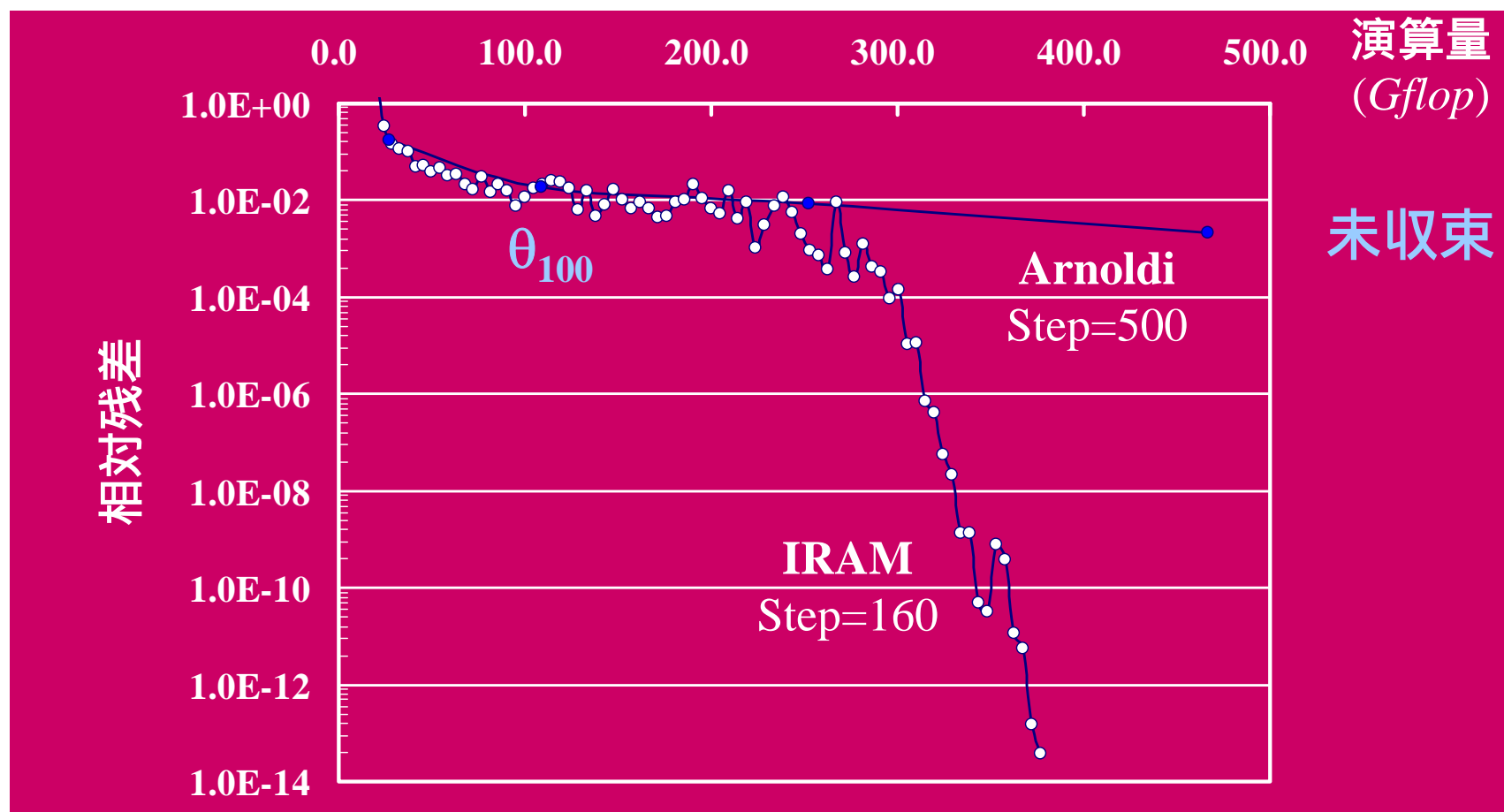
# 性能評価 (3)

- $N=51482$  の収束状況
  - 最大固有値  $\theta_1$  の収束は同等,  $\theta_{30}$  はIRAMが有効



## 性能評価 (4)

- $N=51482$  の収束状況
  - 固有値  $\theta_{100}$  はIRAMは収束, Arnoldiは未収束



# Lanczos 法: 大規模実対称疎行列向け解法

## ■ アイディア

- Arnoldi 法と同様, クリロフ部分空間  $K_k(A; \mathbf{x}^{(0)})$  中で Ritz-Galerkin 法により近似固有値・固有ベクトルを求める
- ただし,  $A$  の対称性を利用し, Lanczos 過程を用いる

## ■ Lanczos 過程

[Lanczos process]

$$s_0 = 0, \mathbf{q}_{-1} = \mathbf{0}, \mathbf{q}_1 = \mathbf{x}^{(0)} / \|\mathbf{x}^{(0)}\|_2$$

for  $k = 1, 2, 3, \dots$

$$\mathbf{v} = A\mathbf{q}_k$$

$$t_k = \mathbf{q}_k^T \mathbf{v}$$

$$\mathbf{v} = \mathbf{v} - s_{k-1}\mathbf{q}_{k-1} - t_k\mathbf{q}_k$$

$$s_k = \|\mathbf{v}\|_2$$

$$\mathbf{q}_{k+1} = \mathbf{v} / s_k$$

end for

} 新たなベクトルの生成

} 直交化

} 正規化

# Lanczos 法 (続き)

- Lanczos 過程の結果
  - 次の式を満たす  $Q_k \in \mathbf{R}^{k \times n}$ ,  $T_k \in \mathbf{R}^{k \times k}$  が得られる

$$Q_k^T A Q_k = T_k.$$

ただし,

$$Q_k = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k], \quad K_k(A; \mathbf{x}^{(0)}) \text{ の正規直交基底}$$

$$T_k = \begin{bmatrix} t_1 & s_1 & & & \\ s_1 & t_2 & \ddots & & \\ & \ddots & \ddots & & \\ & & & s_{k-1} & \\ & & & s_{k-1} & t_k \end{bmatrix}.$$

3重対角行列



# Lanczos 法 (続き)

## ■ アルゴリズムのまとめ

- Lanczos 過程により  $Q_k, T_k$  を生成
- $k \times k$  行列の固有値問題  $T_k \mathbf{y} = \lambda \mathbf{y}$  を解く
  - 小規模固有値問題なので, LAPACK (QR法) などで解く
- 近似固有値として  $\lambda$ , 近似固有ベクトルとして  $\mathbf{x} = Q_k \mathbf{y}$  を採用
  - $\lambda$  をRitz値,  $\mathbf{x}$  をRitzベクトルと呼ぶ

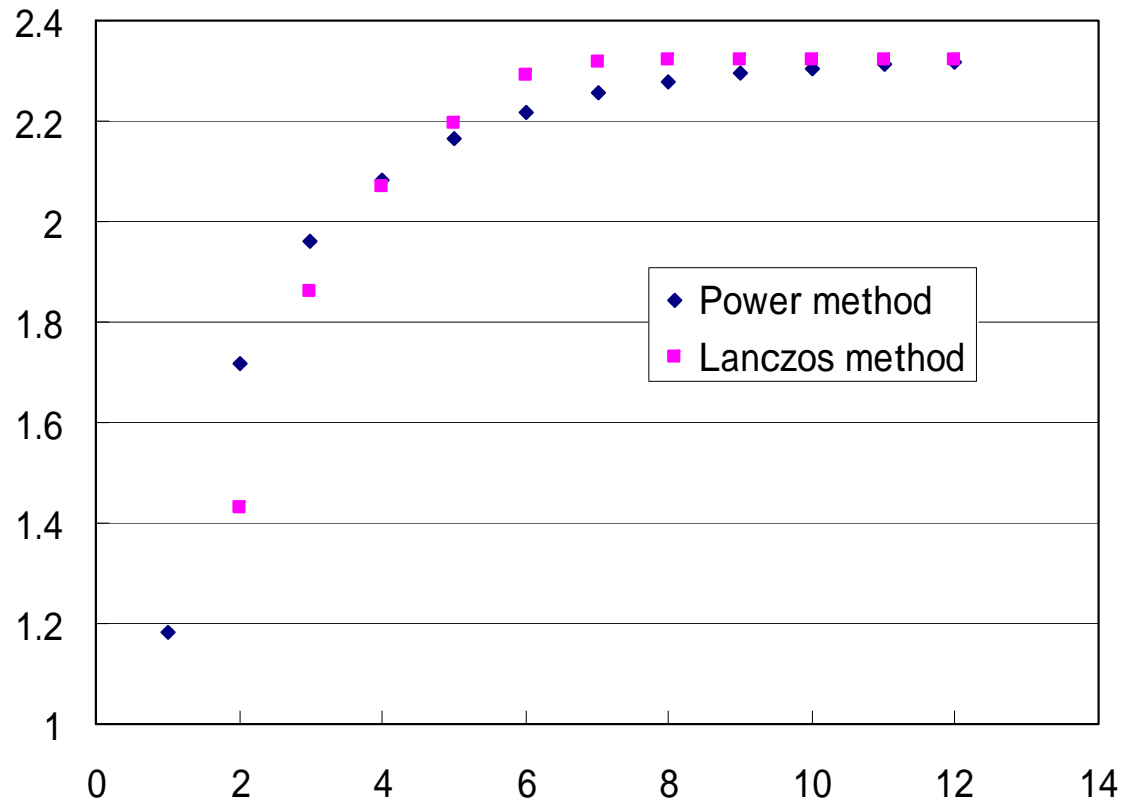
## ■ 演算量

- Lanczos 過程:  $O(kn)$
- $k \times k$  行列の固有値問題:  $O(k^3)$
- 近似固有ベクトルの計算:  $O(kn) \times (\text{必要な固有ベクトル本数})$
- Lanczos 法は通常  $k \ll n$  として使うため, Lanczos 過程の部分が演算量のほとんどを占める

# 数値実験

## ■ べき乗法と Lanczos 法の比較

■  $n = 20$ , 実対称乱数行列







# Lanczos 法：別の見方

## ■ 最大固有値の近似

- 最大固有値は Rayleigh 商  $x^T A x / x^T x$  の最大値であることに着目
- 部分空間  $K_k(A; x^{(0)}) = \text{span}\{x^{(0)}, x^{(1)}, \dots, x^{(k-1)}\}$  中で, これを最大化するベクトルを求める
- $Q_k = [q_1, q_1, \dots, q_k]$  を用いると,

$$\max_{x \in K_n(A; x^{(0)}), \|x\|_2=1} x^T A x = \max_{\|Q_k y\|_2=1} y^T Q_k^T A Q_k y = \max_{\|y\|_2=1} y^T T_k y.$$

- 最右辺は  $T_k$  の最大固有値であり, これを達成する  $y$  は, 最大固有値に属する  $T_k$  の固有ベクトル
- これより,  $A$  の近似固有値, 及び近似固有ベクトル  $x = Q_k y$  が求まる  
これは Lanczos 法の結果と同じ
- すなわち, Lanczos 法では, Rayleigh 商を評価関数として, 部分空間  $\text{span}\{x^{(0)}, x^{(1)}, \dots, x^{(k-1)}\}$  中で  $v_1$  に最も近いベクトルを求めている



# Lanczos 法：別の見方

- 中間固有値の近似

- 大きいほうから  $r$  番目の固有値  $\lambda_r$  を Courant-Fischer の最大・最小定理を使って表現する
- $S$  として,  $K_k(A; \mathbf{x}^{(0)})$  の部分空間のみを考えると,

$$\begin{aligned} & \max_{S: \dim S=r, S \subset K_k(A; \mathbf{x}^{(0)})} \min_{\mathbf{x} \in S, \mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\ &= \max_{P: \dim P=r, P \subset \mathbf{R}^k} \min_{\mathbf{y} \in P, \mathbf{y} \neq \mathbf{0}} \frac{\mathbf{y}^T Q_k^T A Q_k \mathbf{y}}{\mathbf{y}^T Q_k^T Q_k \mathbf{y}} \\ &= \max_{P: \dim P=r, P \subset \mathbf{R}^k} \min_{\mathbf{y} \in P, \mathbf{y} \neq \mathbf{0}} \frac{\mathbf{y}^T T_k \mathbf{y}}{\mathbf{y}^T \mathbf{y}}. \end{aligned}$$

- 最右辺は  $T_k$  の  $r$  番目の固有値だから, Lanczos法の結果に等しい
- すなわち, Lanczos 法は, 部分空間  $K_k(A; \mathbf{x}^{(0)})$  中で Courant-Fischer の最大・最小定理を使うことで, 近似固有値を求めている



# 丸め誤差と再直交化

## ■ 丸め誤差の影響

- Lanczos 法では, ステップが進むにつれ, 丸め誤差のために  $q_1, q_2, \dots, q_k$  が直交しなくなるという現象が生じる
- その結果, **偽固有値**などの問題が生じることがある

## ■ 再直交化

- この問題を解決するには,  $q_1, q_2, \dots, q_k$  を直交化し直せばよい。これを**再直交化**と呼ぶ
- ただし, 全ベクトルの再直交化には  $O(k^2n)$  の演算量が必要であり, 演算量が小さいという Lanczos 法の利点が損なわれてしまう
- そこで, ベクトル間の直交性を評価し, 必要な部分にのみ直交化を行う**選択的直交化**も使われる



# リスタート

---

- ステップ数と演算量

- 再直交化を行う Lanczos 法では, ステップ数  $k$  に対して演算量が  $O(k^2n)$  で増加
- そのため, 適当なところで反復を打ち切り,  $q_k$  を新たな  $x^{(0)}$  として計算を再開するリスタートが用いられる

- リスタートに関する工夫

- $q_k$  をそのまま  $x^{(0)}$  として使うのではなく,  $q_1, q_2, \dots, q_k$  の線形結合を作ってより良いリスタートベクトルを求める Thick Restart Lanczos 法 (TR-Lanczos) \* も提案されている

\* <http://crd-legacy.lbl.gov/~kewu/ps/trlan.html>



# Thick Restart Lanczos 法

## ■ 基本的なアイデア

- Lanczos 法のステップ数に上限値  $k$  を設ける
- 第  $k$  ステップで  $k$  本の Ritz ベクトルを計算
- うち  $l$  本 ( $l < k$ ) を選んで初期ベクトルとし, Lanczos 法をリスタート

## ■ メリット

- リスタートを行っても,  $l$  本分のベクトルの情報が保存される
- 空間の次元の上限が  $k$  に固定      演算量・記憶領域の削減

## ■ 課題と解決法

- Lanczos 分解の形  $AV_k = V_k T_k + \beta_k v_{k+1} e_k^T$  が崩れてしまう
- しかし, Lanczos 法を若干修正したアルゴリズムにより, リスタート後も反復を続けることが可能

# 性能評価 (1)

- 評価環境
  - EP8000 /690Turbo
- 評価例題
  - Matrix Market
- 評価対象の手法
  - Lanczos
  - TR-Lanczos
- 評価指標
  - 収束に必要なベクトル本数
  - 演算量・計算時間
  - 収束状況

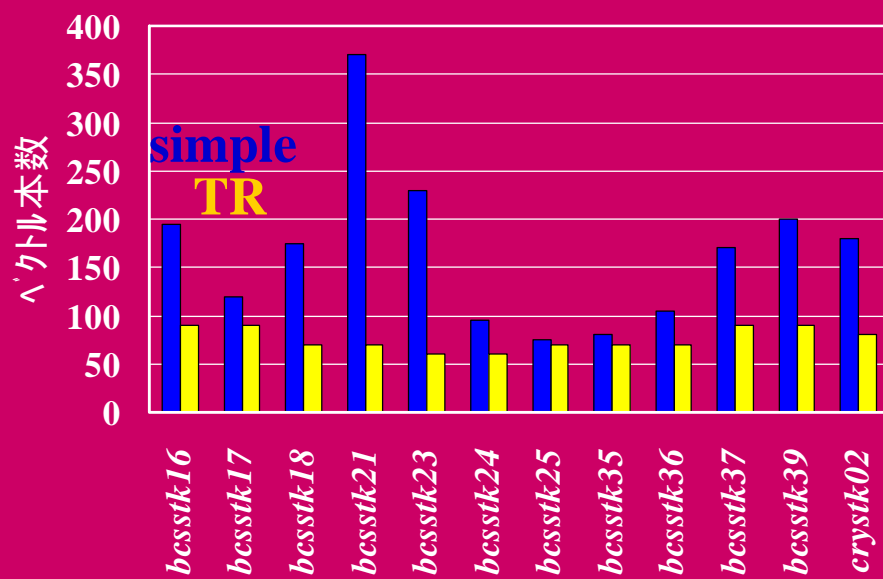
Matrix	$n$	$nz$	Gap
<i>bcsstk16</i>	4884	147631	0.58
<i>bcsstk17</i>	10974	219812	0.88
<i>bcsstk18</i>	11984	80519	0.73
<i>bcsstk21</i>	3600	15100	<b>0.13</b>
<i>bcsstk23</i>	3134	24156	0.98
<i>bcsstk24</i>	3562	81736	1.00
<i>bcsstk25</i>	15439	133840	1.00
<i>bcsstk35</i>	30237	740200	0.99
<i>bcsstk36</i>	23052	583096	0.82
<i>bcsstk37</i>	25503	583240	0.60
<i>bcsstk39</i>	46772	1068034	<b>0.34</b>
<i>crystk02</i>	13965	491274	0.99

密集度 **Gap** =  $(\lambda_1 - \lambda_{100}) / \lambda_1$

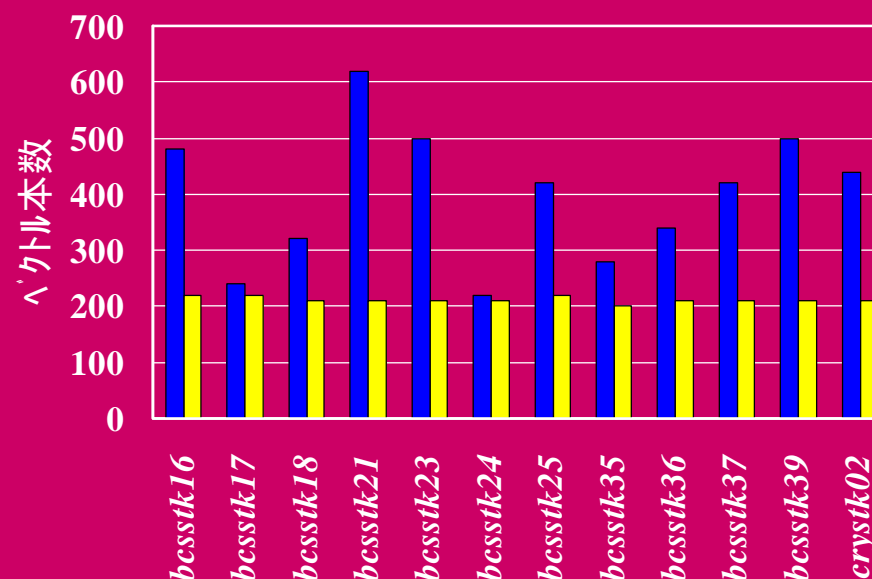
## 性能評価 (2)

- 収束判定基準  $\|Ax - x\theta\| / \|\theta\| < 10^{-10}$  (相対残差)
- 所望固有値個数  $N_E = 30, 100$
- 結果 **TR: 所要ベクトル本数少ない**  
**密集固有値に対しても安定**

### 収束に必要なベクトル本数の比較



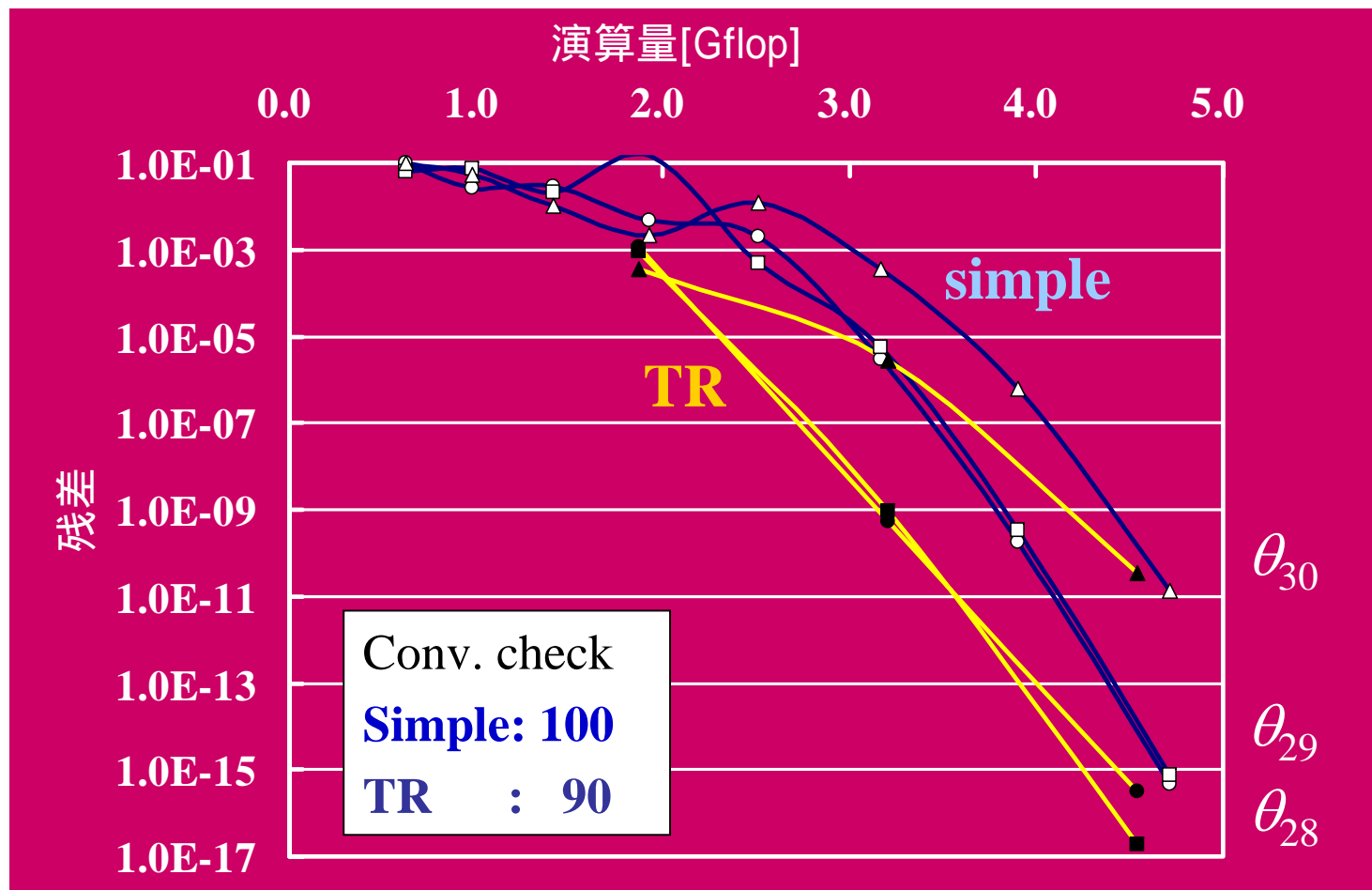
$N_E=30$



$N_E=100$

## 性能評価 (3)

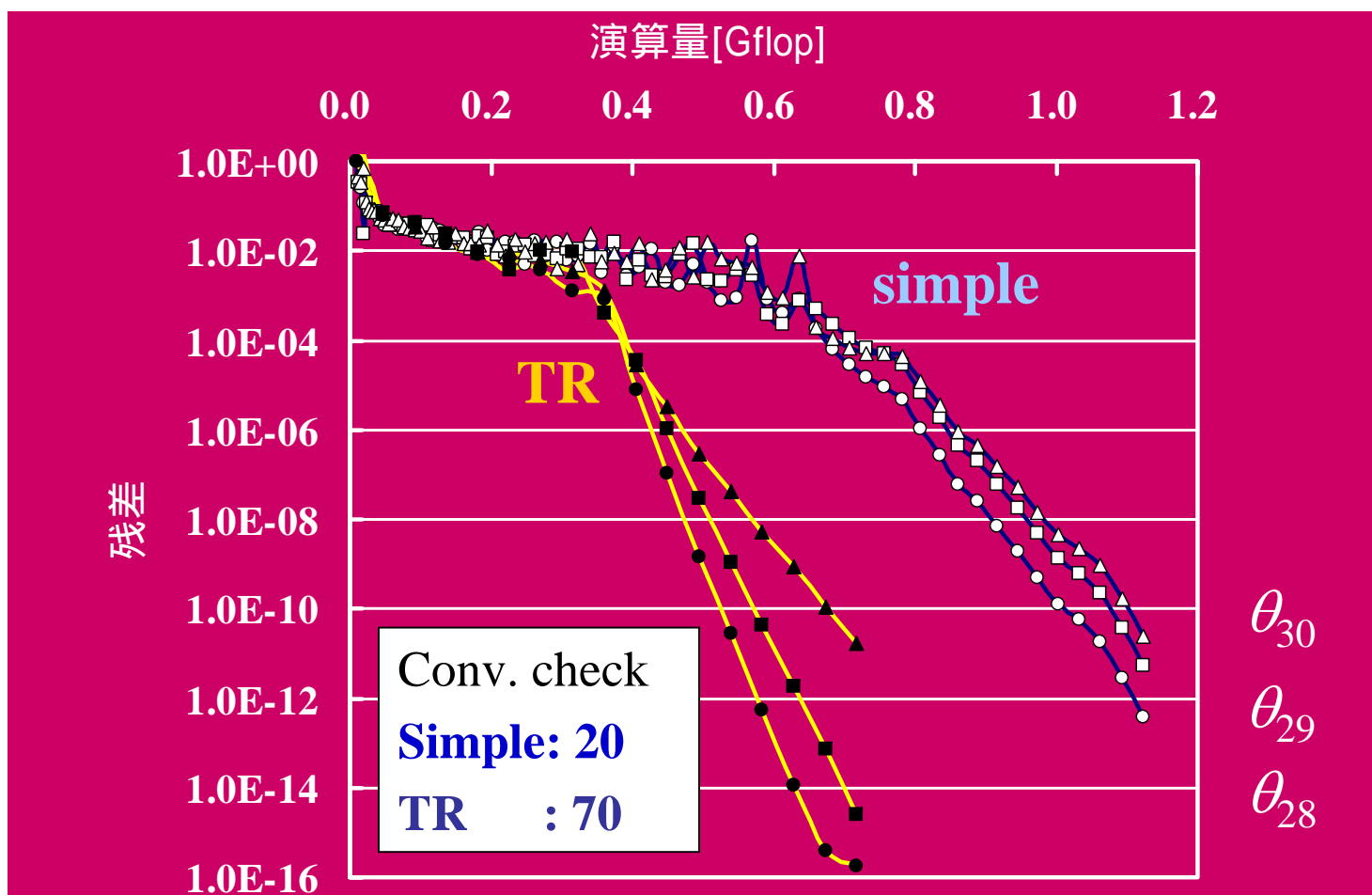
- 密集固有値がない場合: *bcsstk39*
- 最大固有値から28, 29, 30個の固有値: 収束性は同等





## 性能評価 (4)

- 密集固有値がある場合: *bcsstk21*
- TR-Lanczosは密集固有値に有効





## 終わりに

---

- 本発表では、大規模疎行列に対する固有値解法として、非対称行列向けの Arnoldi 法と対称行列向けの Lanczos 法を説明した
- これらのアルゴリズムでは、演算量・記憶領域の増大と丸め誤差蓄積を避けるため、リスタートが重要である。効率的にリスタートを行う方法として、Implicitly Restarted Arnoldi法と Thick Restart Lanczos法がある
- 今回は紹介しなかったが、中間固有値問題には、Jacobi-Davidson法も有効
- IRAM, TRLAN, JD にはフリーのソースコードも用意されているので、解きたい問題の特徴に合った解法を選んで実際に試してみるのがよい



## 参考文献

---

- L. N. Trefethen and D. Bau III: “Numerical Linear Algebra”, SIAM, Philadelphia, 1997.
- Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. Van der Vorst (eds.): “Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide”, SIAM, Philadelphia, 2000.
- G. W. Stewart: “Matrix Algorithms, Vol. II: Eigensystems”, SIAM, Philadelphia, 2001.
- Y. Saad: “Iterative Methods for Sparse Linear Systems”, SIAM, Philadelphia, 2011.
- 杉原正顯, 室田一雄: “線形計算の数理”, 岩波書店, 2009.