

固有値計算アルゴリズムの最近の進展

II. 疎行列向け解法

2003年5月14日

山本有作

目次

1. 疎行列向け固有値計算の概要
2. 射影法
3. Arnoldi法
4. Lanczos法
5. Jacobi-Davidson法

1. 疎行列向け固有値計算の概要

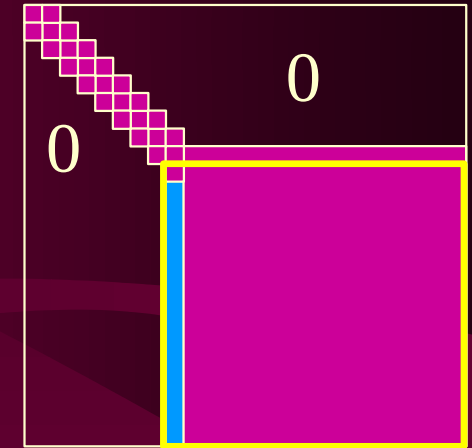
- 対象とする問題
- 疎行列固有値問題の特徴
- 疎行列固有値問題のための反復解法

対象とする問題

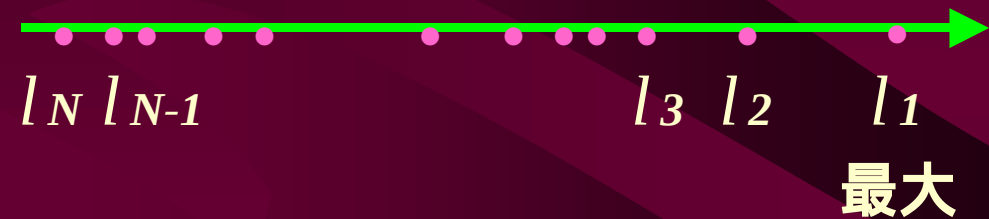
- 標準固有値問題 $A\mathbf{x} = \lambda \mathbf{x}$
 - A : 対称(エルミート)または非対称疎行列
 - A のサイズは数万～数億次元
- 一般固有値問題 $A\mathbf{x} = \lambda M\mathbf{x}$
 - M : 対称疎行列
 - M は特異の場合もあり。
 - 有限要素法, 電子状態計算などで直交しない基底関数を用いた場合に現れる。
- 応用分野
 - 構造解析, 電子状態計算, 導波路解析, 文書検索など

疎行列固有値問題の特徴

- 行列サイズが巨大
 - 密行列形式で主記憶に格納することが不可能
 - 行列を変形していく Householder 法は適用不可
- 必要な固有値・固有ベクトルは少数のみ
 - 最大・最小付近の固有値を求めたい場合が多い。



反復解法の利用



- 反復解法の特徴
 - 行列ベクトル積という形でのみ行列データを使用
 - 最大・最小付近の固有値・固有ベクトルのみを効率良く計算可能

疎行列固有値計算のための反復解法

- Lanczos法 (Lanczos, 1950)
 - 対称疎行列向け
 - 行列を三重対角行列に変換
 - 精度・安定性・収束性向上のための多数の変種あり
- Arnoldi法 (Arnoldi, 1951)
 - 非対称疎行列向け
 - 行列を近似的に Hessenberg 行列に変換
 - 精度・安定性・収束性向上のための多数の変種あり
- Jacobi-Davidson法 (Sleijpen & van der Vorst, 1996)
 - 対称 / 非対称疎行列向け

2. 射影法

- 射影法の原理
- Ritz 値と Ritz ベクトル
- 射影法に基づく反復解法
- 収束性の向上

射影法の原理

- 基本的なアイデア
 - 部分空間への射影により, N 元の大規模固有値問題 $A\mathbf{x} = \lambda \mathbf{x}$ を小さな固有値問題で近似
- 部分空間への射影
 - 部分空間 $S_m = \text{span} \{ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m \}$ を設定
($\{\mathbf{v}_i\}$: 正規直交基底, $m \ll N$)
 - $A\mathbf{x} = \theta \mathbf{x}$ を満たす $\mathbf{x} \in S_m$, θ を求めたいが, これは一般に不可能
 - 代わりに,

$$\langle \mathbf{w}, A\mathbf{x} - \theta \mathbf{x} \rangle = 0 \quad \text{for} \quad \mathbf{w} \in S_m$$

(残差 $A\mathbf{x} - \theta \mathbf{x}$ の S_m 方向の成分が0)

 を満たす $\mathbf{x} \in S_m$, θ を探す。

Ritz 値と Ritz ベクトル

- 小さな固有値問題への変換

- $V_m = [v_1, v_2, \dots, v_m], x = V_m s$ とおくと,

$$\langle w, Ax - \theta x \rangle = 0 \quad \text{for } w \in S_m$$

$$V_m^t (A V_m s - \theta V_m s) = 0$$

$$(V_m^t A V_m) s = \theta s$$

- k 元の固有値問題 $(V_m^t A V_m) s = \theta s$ を解けば, x と θ が求まる。
 - θ を **Ritz 値**, s を **Ritz ベクトル** と呼ぶ。

射影法に基づく反復解法

- 部分空間の逐次的拡大

初期ベクトル $\mathbf{v}_1$ を設定

$$V_1 = [\mathbf{v}_1]$$

DO $i = 2, m$

部分空間拡大のための新しいベクトル $\mathbf{u}$ を求める。

$\mathbf{u}$ を V_{i-1} に対して正規直交化し, $\mathbf{v}_i$ を求める。

$$V_i = [V_{i-1} \mid \mathbf{v}_i]$$

END DO

$(V_m^t A V_m) \mathbf{s} = \theta \mathbf{s}$ を解いて $\theta, \mathbf{s}$ を求める。

$\theta, \mathbf{x} = V_m \mathbf{s}$ をそれぞれ A の固有値, 固有ベクトルとして採用

- 部分空間を拡大するにつれ, 解の精度は向上
- Lanczos 法, Arnoldi 法, JD 法はいずれもこの範疇に入る。

収束性の向上

- 指導原理

- 求めたい固有ベクトルの成分が多く含まれるように, 部分空間 S_m を選ぶ。



- 実現方法

- (1) 良い初期ベクトル v_1 を選ぶ。
 - Implicit Restart Arnoldi 法
 - Thick Restart Lanczos 法
- (2) 部分空間拡大のためのベクトル u をうまく選ぶ。
 - Jacobi-Davidson 法

3. Arnoldi 法

- Arnoldi 法の原理
- アルゴリズム
- 従来の Arnoldi 法の問題点
- Implicit Restart Arnoldi 法
- 一般固有値問題への適用
- 性能評価

Arnoldi 法の原理 (1)

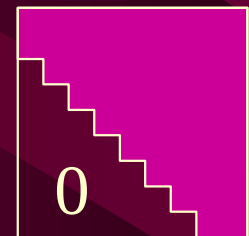
- 部分空間の設定
 - $\mathbf{v}_1$ は任意に選ぶ。
 - $S_m = K_m(A, \mathbf{v}_1) = \text{span} \{ \mathbf{v}_1, A\mathbf{v}_1, A^2\mathbf{v}_1, \dots, A^{m-1}\mathbf{v}_1 \}$
(Krylov 部分空間)
- Krylov 部分空間を選ぶメリット
 - m を増やすに連れ、絶対値最大の固有値に対する固有ベクトルの成分が増幅される。
 - したがって、絶対値最大の固有値に対する固有ベクトルが効率良く近似できる。

Arnoldi 法の原理 (2)

- 正規直交基底の生成
 - 第 j ステップでは, $\mathbf{v}_j$ に A をかけ, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j$ に対して直交化することにより, 新しい基底ベクトル $\mathbf{v}_{j+1}$ を生成
- Arnoldi 分解
 - $\mathbf{v}_{j+1}$ の作り方より, $A\mathbf{v}_j$ は $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{j+1}$ の線形結合
 - したがって, $V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$ は次の式を満たす。

$$AV_m = V_m H_m + h_{m+1, m} \mathbf{v}_{m+1} \mathbf{e}_m^t$$

ただし, H_m : $m \times m$ の Hessenberg 行列
 $h_{m+1, m}$: スカラー値
 $\mathbf{e}_m$: 第 m 成分のみが1の m 次元ベクトル



Hessenberg行列

これを A の **m-step Arnoldi 分解**と呼ぶ。

初期ベクトル $\mathbf{v}_1$ を設定

$$\mathbf{v}_1 := \mathbf{v}_1 / \|\mathbf{v}_1\|_2$$

$$V_0 = \varnothing$$

DO $j = 1, m$

$$V_j = [V_{j-1} \mid \mathbf{v}_j]$$

$$\mathbf{u} = A\mathbf{v}_j$$

DO $i = 1, j$

$$h_{ij} = \mathbf{v}_i^t \mathbf{u}$$

$$\mathbf{u} := \mathbf{u} - h_{ij} \mathbf{v}_i$$

END DO

$$h_{j+1,j} = \|\mathbf{u}\|_2$$

$$\mathbf{v}_{j+1} = \mathbf{u} / h_{j+1,j}$$

END DO

$$H_m = (h_{ij})_{i,j=1}^m$$

$H_m \mathbf{s} = \theta \mathbf{s}$ を解いて $\theta, \mathbf{s}$ を求める。

$\theta, \mathbf{x} = V_m \mathbf{s}$ をそれぞれ A の固有値, 固有ベクトルとして採用

A の乗算による部分空間拡大

Modified Gram-Schmidt 法による直交化

収束判定

- 残差の計算

- $H_m s = \theta s$ が成り立つとき, $x = V_m s$ とすると

$$\begin{aligned} Ax - x \theta &= AV_m s - V_m H_m s \\ &= h_{m+1, m} v_{m+1} e_m^t s \\ &= h_{m+1, m} e_m^t s \end{aligned}$$

- したがって, $E = h_{m+1, m} (e_m^t s) v_{m+1} x^t$ とおくと, x と θ は

$$(A + E) x = \theta x$$

を満たす。即ち, A に摂動 E を加えた固有値問題の解となる。



$h_{m+1, m} e_m^t s$ の大きさにより, 収束判定を行う。

Arnoldi 法の問題点

第 i ステップでは, $v_1, v_2, \dots, v_i$ に対する直交化が必要



第 i ステップでのメモリ所要量, 演算量は i に比例して増大
ステップ数 m に対し, メモリ所要量は $O(m)$, 演算量は $O(m^2)$ で増大



リスタートによるメモリ所要量・演算量削減

従来のリスタート法とその問題点

- 従来のリスタート法
 - 反復回数 m に上限値を設定
 - 上限値まで反復しても収束しなければ, $\mathbf{v}_m$ を $\mathbf{v}_1$ として, 再び最初から Arnoldi 法を実行
- 問題点
 - ベクトル $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m-1}$ 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。

Implicit Restart Arnoldi 法 (1)

- 基本的なアイデア
 - リスタート時に $\mathbf{v}_m$ だけでなく, 部分空間 S_m 中の k 本 ($k < m$) のベクトルを保存する。
 - k 本のベクトルは, 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, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ の線形結合として選ぶ。
- メリット
 - リスタートを行っても, k 本分のベクトルの情報が保存される。
 - 基底ベクトルの本数の上限が m に固定される。
 - メモリ所要量の削減
 - 直交化の演算量削減

- 実現方法

- m-step Arnoldi 分解 $AV_m = V_m H_m + h_{m+1, m} \mathbf{v}_{m+1} \mathbf{e}_m^t$ が入力
- シフト $\mu_1, \mu_2, \dots, \mu_{m-k}$ を用い, H_m に対して $m-k$ ステップのQR法を実行すると, 次の新しい関係式を得る。

$$AV_m^+ = V_m^+ H_m^+ + h_{m+1, m} \mathbf{v}_{m+1} \mathbf{e}_m^t Q$$

ただし $V_m^+ = V_m Q, \quad H_m^+ = Q^t H_m Q,$

$$Q = Q_1 Q_2 \dots Q_{m-1} \quad (\text{QR法の直交変換の行列})$$

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

$$AV_k^+ = V_k^+ H_k^+ + h_{k+1, k} \mathbf{v}_{k+1}^+ \mathbf{e}_k^t$$

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

- シフトと初期ベクトルの関係

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

$$QR = (H_m^+ - \mu_1) (H_m^+ - \mu_2) \dots (H_m^+ - \mu_{m-k})$$

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

- したがって, V_m^+ の第1列を $\mathbf{v}_1^+$ とすると,

$$\begin{aligned} \mathbf{v}_1^+ &= V_m Q \mathbf{e}_1 \\ &= V_m (H_m^+ - \mu_1) (H_m^+ - \mu_2) \dots (H_m^+ - \mu_{m-k}) R^{-1} \mathbf{e}_1 \\ &= r_{11}^{-1} (A - \mu_1) (A - \mu_2) \dots (A - \mu_{m-k}) \mathbf{v}_1 \end{aligned}$$

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

$$\mathbf{v}_1 \rightarrow (A - \mu_1) (A - \mu_2) \dots (A - \mu_{m-k}) \mathbf{v}_1$$

と変えたことと等価となる。

Implicit Restart Arnoldi 法 (5)

- シフトの選び方
 - μ をシフトとして選ぶと, 初期ベクトル v_1 のうち, μ 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**の固有値に近い値に選ぶ。

Arnoldi 分解 $AV_m = V_m H_m + h_{m+1, m} \mathbf{v}_{m+1} \mathbf{e}_m^t$ を入力。

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

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

$$Q = Q_1 Q_2 \dots Q_{m-1}$$

$$V_m^+ = V_m Q, \quad H_m^+ = Q^t H_m Q$$

$$\beta_k = (H_m^+)_{k+1, k}, \quad \sigma_k = Q_{m, k}$$

$$\mathbf{v}_{k+1}^+ = \mathbf{v}_{k+1} \beta_k + h_{m+1, m} \mathbf{v}_{m+1} \sigma_k$$

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

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

V_m^+ の最初の第 k 列からなる行列を V_k^+ とする。

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

Arnoldi 分解 $AV_k^+ = V_k^+ H_k^+ + h_{k+1, k} \mathbf{v}_{k+1}^+ \mathbf{e}_k^t$ から始めて, Arnoldi法を $m-k$ ステップ行う。

END DO

一般固有値問題への適用 (1)

- 対象とする問題

- $A\mathbf{x} = \lambda M\mathbf{x}$
- M は一般に対称疎行列
- M は特異の場合もあり。

- 解法 I: 標準固有値問題に直す方法

- $M = LL^t$ と Cholesky 分解を行い、 $(L^{-1}AL^{-t})(L^t\mathbf{x}) = \lambda (L^t\mathbf{x})$ を解く。
- 問題点
 - fill-in により、 L の非零要素数が M に比べて大きく増加
 - M が特異の場合に対応不可

一般固有値問題への適用 (2)

26

- 解法 II: スペクトル変換を使う方法
 - 同値変形

$$\begin{aligned} A\mathbf{x} &= \lambda M\mathbf{x} & (A - \sigma M)\mathbf{x} &= (\lambda - \sigma) M\mathbf{x} \\ & & (A - \sigma M)^{-1}M\mathbf{x} &= \nu \mathbf{x} \\ \text{ただし } \nu &= 1 / (\lambda - \sigma) \end{aligned}$$

により, 標準固有値問題に変換

- 利点
 - σ の調節により, 内部固有値も効率よく求められる。
 - M が特異であっても対応可能
- 問題点
 - $(A - \sigma M)^{-1}$ による乗算が必要

一般固有値問題への適用 (3)

- 解法 III: M -内積に基づく方法
 - Arnoldi 法において, 内積 $\mathbf{v}^t \mathbf{u}$, ノルム $\|\mathbf{u}\|_2$ をすべて, M -内積 $\mathbf{v}^t M \mathbf{u}$, M -ノルム $\sqrt{\mathbf{u}^t M \mathbf{u}}$ に置き換える。



$A\mathbf{x} = \lambda M\mathbf{x}$ を満たし, M -直交性を持つ固有ベクトルが得られる。

- 解法 II と組み合わせて使うことも可能

性能評価 () ~ *Arnoldi*法

28

- 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

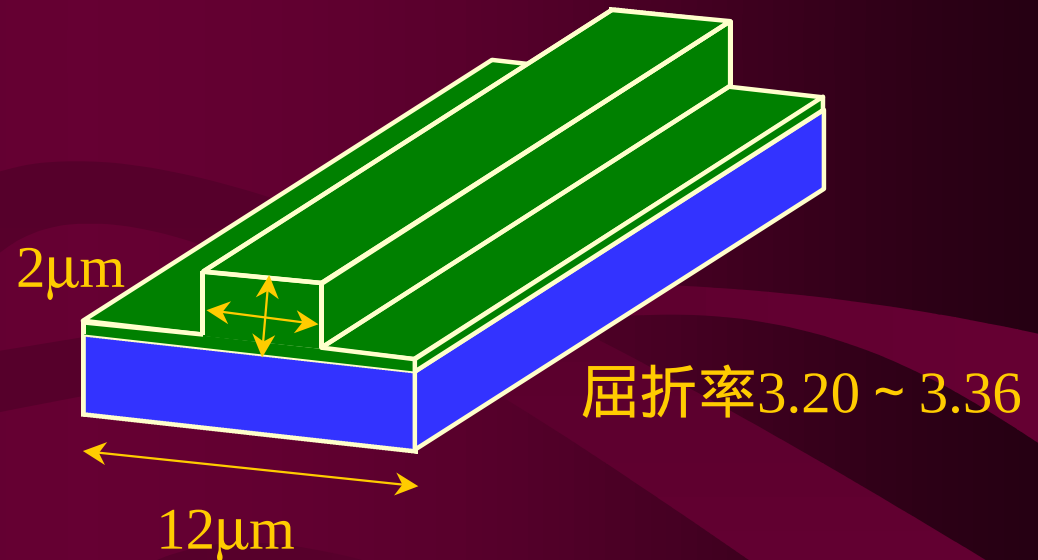
- 評価対象の手法

Arnoldi vs. IRAM

(ARPACK)

収束に必要な演算量

収束状況



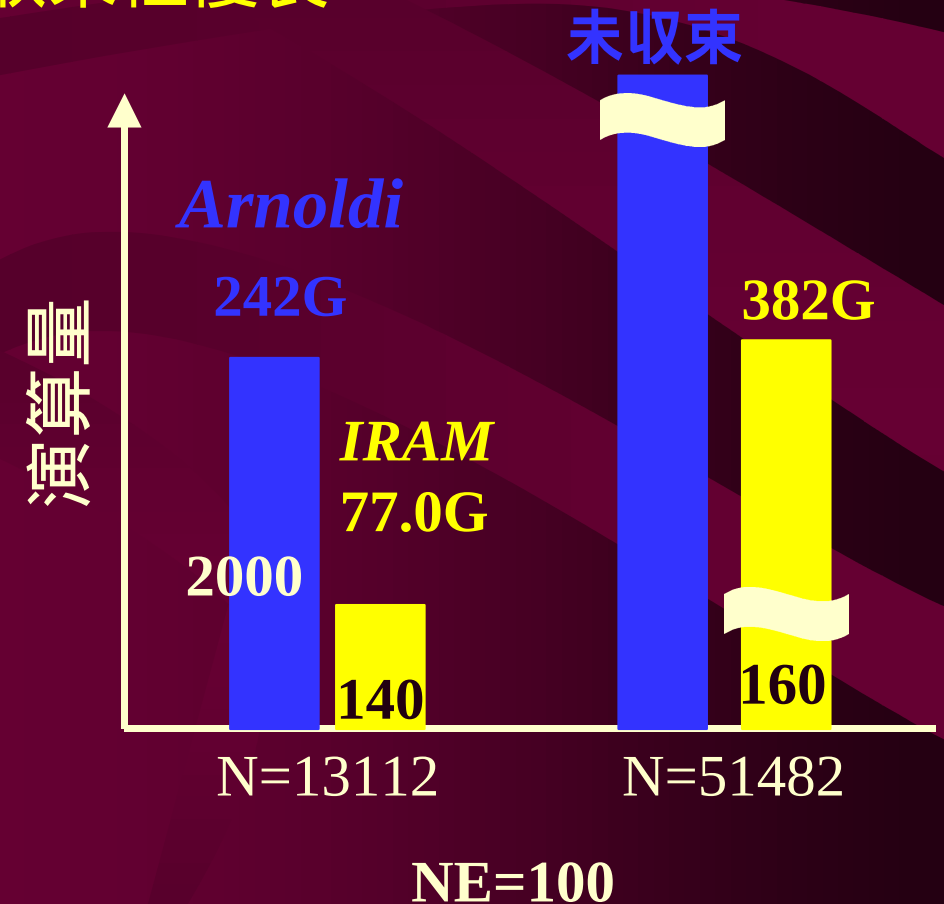
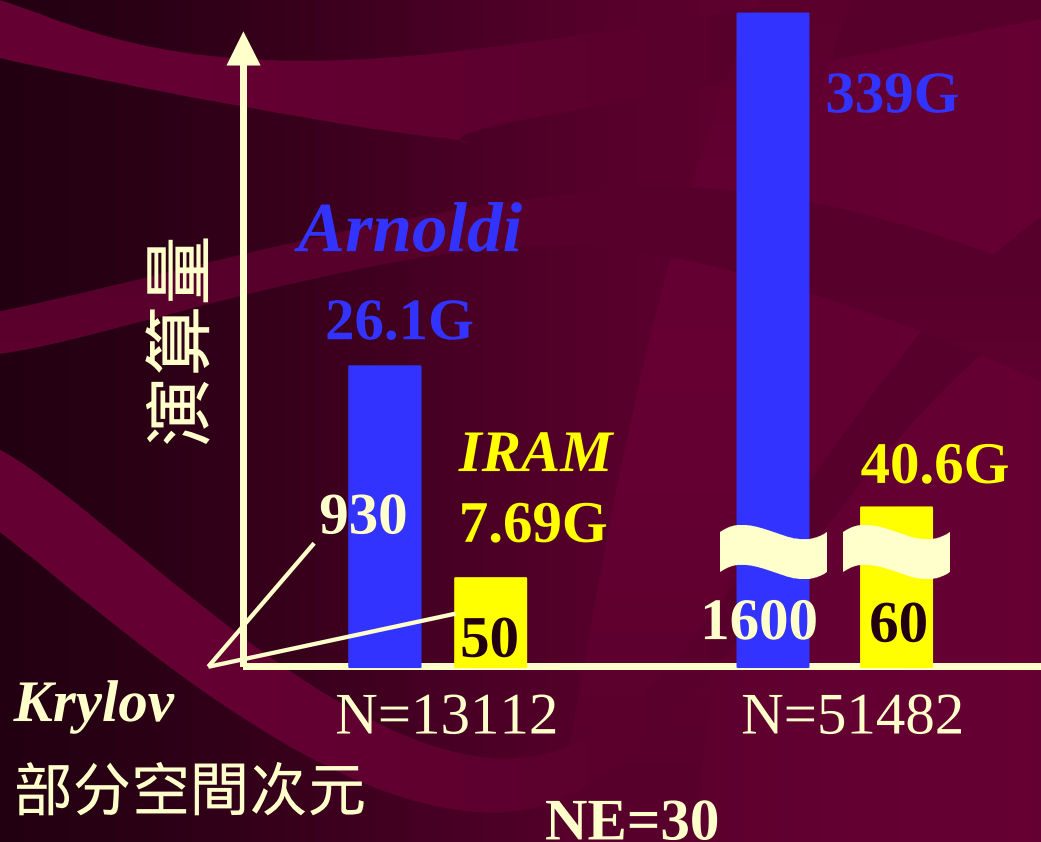
$$\nabla^2 \mathbf{E}(\mathbf{r}) + k_0^2 \epsilon_r \mathbf{E}(\mathbf{r}) = -\nabla [\nabla \log(\epsilon_r) \cdot \mathbf{E}(\mathbf{r})]$$

$$\mathbf{E}(\mathbf{r}) = \begin{bmatrix} E_x \\ E_y \end{bmatrix} \exp(-ibz)$$

連立非対称固有値問題

性能評価 () ~Arnoldi法

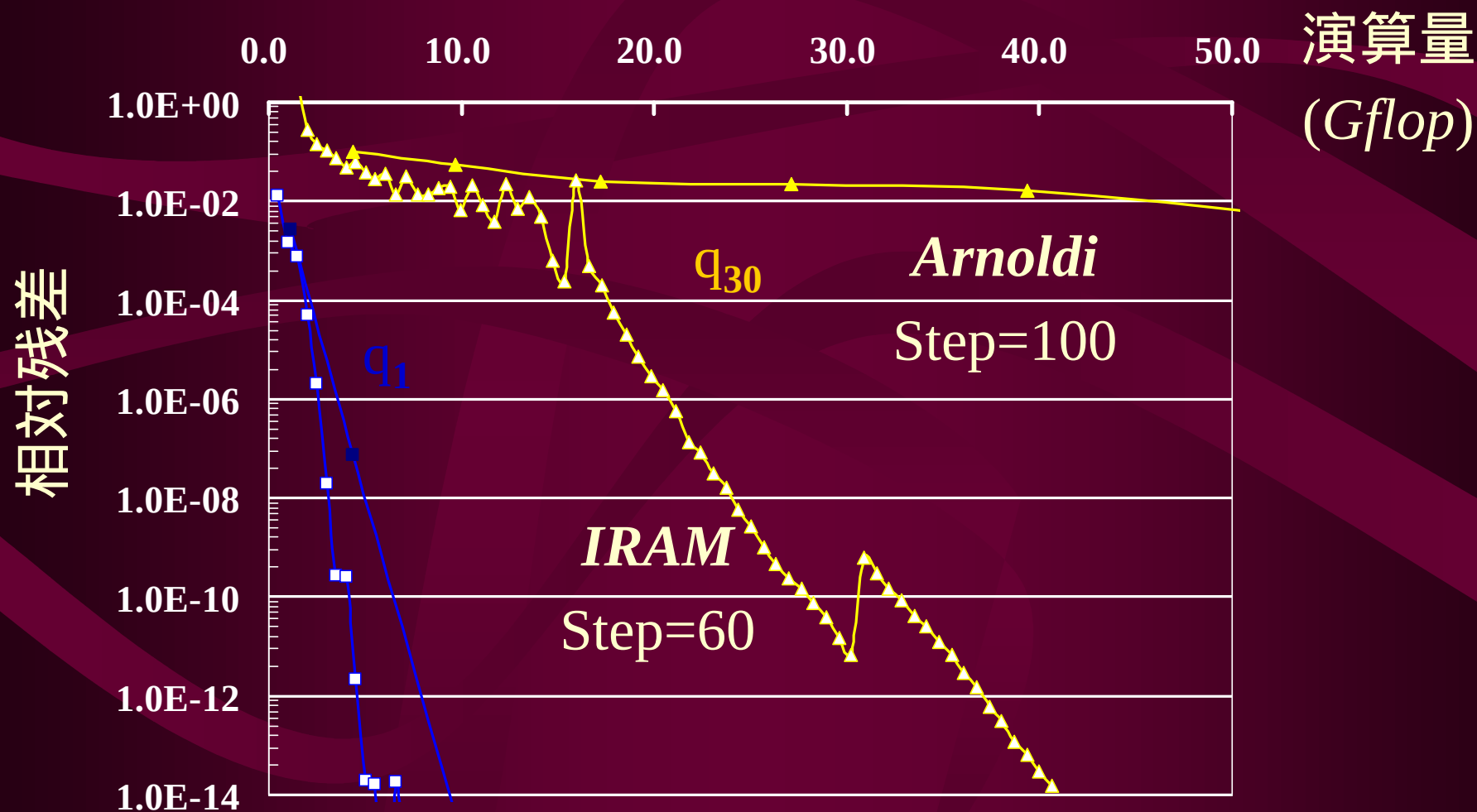
- 収束判定基準 $\|Ax - x\| / \|x\| < 10^{-14}$ 相対残差
- 行列サイズ $N = 13112, 51482$
- 所望固有値個数 $NE = 30, 100$
- 結果 **IRAMは収束性優良**



性能評価 () ~ *Arnoldi*法 収束性(1)

- $n=51482$ の収束状況

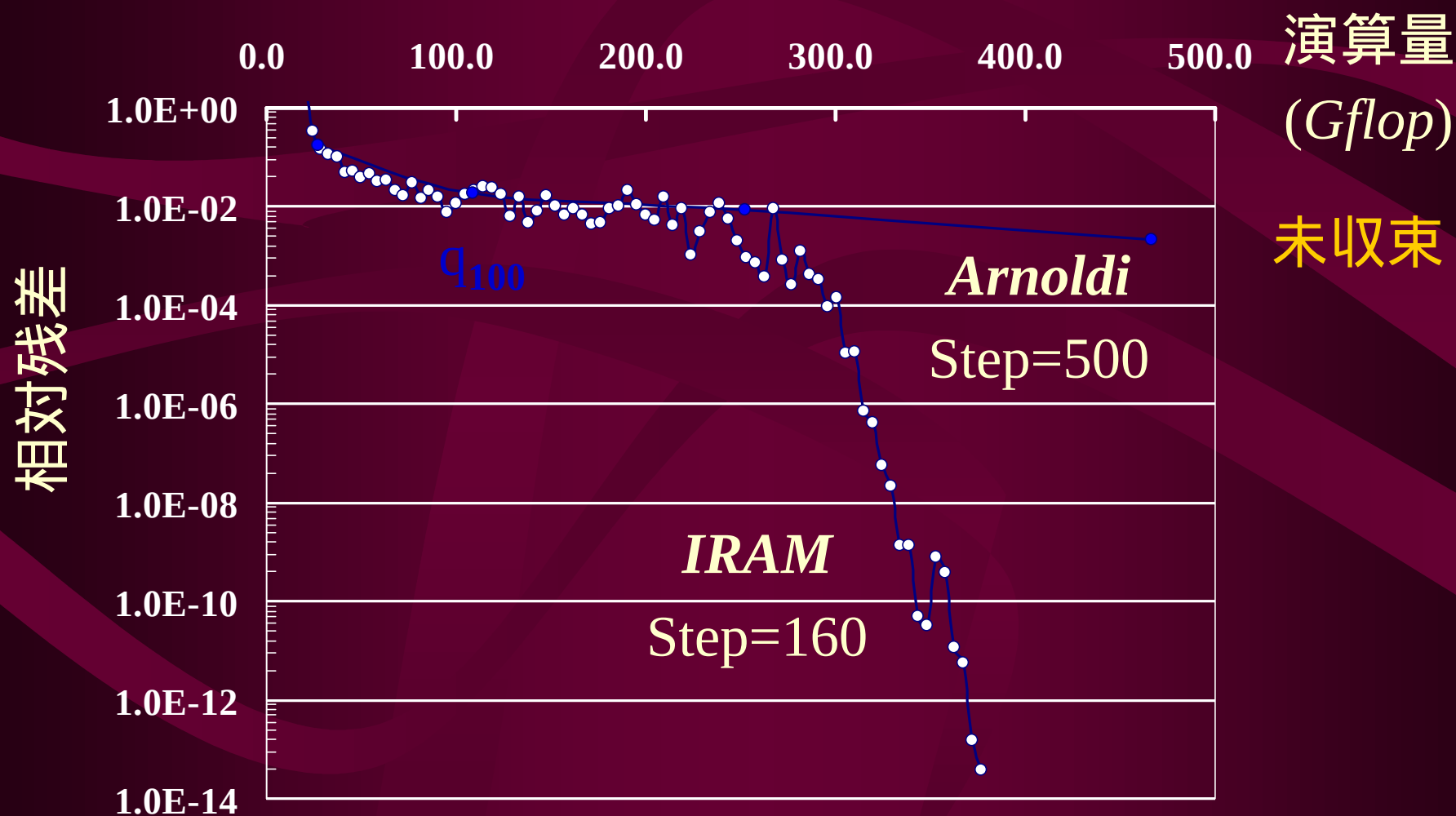
最大固有値 θ_1 の収束は同等, θ_{30} は*IRAM*が有効



性能評価 () ~ *Arnoldi*法

- N=51482 の収束状況

固有値 θ_{100} は *IRAM* は収束, *Arnoldi* は未収束



4. Lanczos 法

- Lanczos 法の原理
- アルゴリズム
- 収束定理
- 従来のLanczos 法の問題点
- ブロック化 Lanczos 法
- Thick Restart Lanczos 法
- 性能評価

Lanczos 法の原理 (1)

- 部分空間の設定

- Arnoldi 法と同様, S_m として Krylov 部分空間を使用
- $S_m = K_m(A, \mathbf{v}_1) = \text{span} \{ \mathbf{v}_1, A\mathbf{v}_1, A^2\mathbf{v}_1, \dots, A^{m-1}\mathbf{v}_1 \}$

- 正規直交基底の生成

- 第 j ステップでは, $\mathbf{v}_j$ に A をかけ, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j$ に対して直交化することにより, 新しい基底ベクトル $\mathbf{v}_{j+1}$ を生成
- $j < m - 1$ のとき

$$\begin{aligned} (A\mathbf{v}_m)^t \mathbf{v}_j &= \mathbf{v}_m^t (A\mathbf{v}_j) \\ &= \mathbf{v}_m^t \sum_{i=1}^{j+1} c_i \mathbf{v}_i \\ &= \sum_{i=1}^{j+1} c_i \mathbf{v}_m^t \mathbf{v}_i = 0 \end{aligned}$$

- したがって, 新たに生成したベクトル $A\mathbf{v}_m$ は $\mathbf{v}_{m-1}, \mathbf{v}_m$ のみに対して直交化すればよい。

- Lanczos 分解

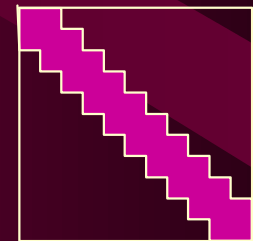
- Lanczos 法は Arnoldi 法の対称行列への適用であるから, 次の Arnoldi 分解の式が成り立つ。

$$AV_m = V_m H_m + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^t$$

- ここで, 両辺に左から V_m^t をかけると, 基底ベクトルの直交性より $V_m^t AV_m = H_m$ 。左辺は対称行列であるから, H_m は Hessenberg かつ対称行列で, すなわち三重対角行列 T_m となる。したがって

$$AV_m = V_m T_m + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^t$$

これを A の **m-step Lanczos 分解**と呼ぶ。



三重対角行列

初期ベクトル $\mathbf{v}_1$ を設定

$$\beta_0 = 0, \quad \mathbf{v}_0 = \mathbf{0}, \quad \mathbf{v}_1 := \mathbf{v}_1 / \|\mathbf{v}_1\|_2$$

$$V_0 = \varphi$$

DO $j = 1, m$

$$V_j = [V_{j-1} \mid \mathbf{v}_j]$$

$$\mathbf{u} = A\mathbf{v}_j - \beta_{j-1} \mathbf{v}_{j-1}$$

$$\alpha_j = \mathbf{v}_j^t \mathbf{u}$$

$$\mathbf{u} := \mathbf{u} - \alpha_j \mathbf{v}_j$$

$$\beta_j = \|\mathbf{u}\|_2$$

$$\mathbf{v}_{j+1} = \mathbf{u} / \beta_j$$

END DO

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

Aの乗算による部分空間拡大
直交化

$T_m \mathbf{s} = \theta \mathbf{s}$ を解いて $\theta, \mathbf{s}$ を求める。

$\theta, \mathbf{x} = V_m \mathbf{s}$ をそれぞれAの固有値, 固有ベクトルとして採用

- 残差の計算

- $T_m s = \theta s$ が成り立つとき, $x = V_m s$ とすると

$$\begin{aligned} Ax - x\theta &= AV_m s - V_m T_m s \\ &= \beta_{m+1} v_{m+1} e_m^t s \\ &= \beta_{m+1} e_m^t s \end{aligned}$$

- したがって, $E = \beta_{m+1} (e_m^t s) v_{m+1} x^t$ とおくと, x と θ は

$$(A + E)x = \theta x$$

を満たす。即ち, A に摂動 E を加えた固有値問題の解となる。
このとき, $|\theta - \lambda_i| < \|E\|_2$ を満たす A の固有値 λ_i が存在する。



$\beta_{m+1} e_m^t s$ の大きさにより, 収束判定を行う。

収束定理

対称行列Aの固有値を $\lambda_1 > \lambda_2 > \dots > \lambda_N$ とし, 初期ベクトル $\mathbf{v}_1$ が固有ベクトル $\{\mathbf{x}_i\}_{i=1}^N$ を用いて

$$\mathbf{v}_1 = \sum_{i=1}^N c_i \mathbf{x}_i$$

と展開されたとする。

このとき, m ステップの Lanczos 法で求めた λ_1 の近似値を $\lambda_1^{(m)}$ とすると,

$$0 \leq \lambda_1 - \lambda_1^{(m)} \leq (\lambda_1 - \lambda_N) \left(\sum_{i=2}^N |c_i|^2 \right) / |c_1|^2 \cdot 4 \left(\sqrt{(1+\gamma)} + \sqrt{\gamma} \right)^{-4(m-1)}$$

ただし, $\gamma = (\lambda_1 - \lambda_2) / (\lambda_1 - \lambda_N)$ 。

Lanczos 法の問題点 I: 再直交化

- 再直交化の必要性
 - 数学的には, Av_j は v_{j-1} , v_j のみに対して直交化すればよい。
 - しかし有限精度での計算では, 丸め誤差の影響により, v_{j-2} 以前のベクトルとの直交性も崩れる場合がある。



精度を保つため, 再直交化が必要



計算量が増大し, Lanczos 法のメリットが減少

再直交化のための従来手法

- Selective orthogonalization (Parlett & Scott, 1979)
 - 直交性の崩れは, 主に収束しつつある Ritz ベクトルの方向に対して起きる。
 - 新たに求めたベクトル $A\mathbf{v}_j$ を, この Ritz ベクトルのみに対して再直交化
- Partial orthogonalization (Simon, 1984)
 - 直交性の崩れ $\omega_{ij} = \mathbf{v}_i^t \mathbf{v}_j$ の満たす方程式を求める。
 ω_{ij} が大きくなったときに限り, 再直交化を行う。

Lanczos 法の問題点II: 縮重固有値のある場合

- 縮重固有値のある場合の Krylov 部分空間の拡大
 - $\mathbf{v}_1 = \sum_{i=1}^N c_i \mathbf{x}_i$ とすると, $A^{j-1} \mathbf{v}_1 = \sum_{i=1}^N \lambda_i^{j-1} c_i \mathbf{x}_i$
 - もし $\lambda_i = \lambda_{i'}$ だとすると, $\mathbf{x}_i$ と $\mathbf{x}_{i'}$ の係数の比は常に $c_i : c_{i'}$



m を増やしても, この縮重固有値に対する部分空間は, 常に1次元のままで拡大されない。

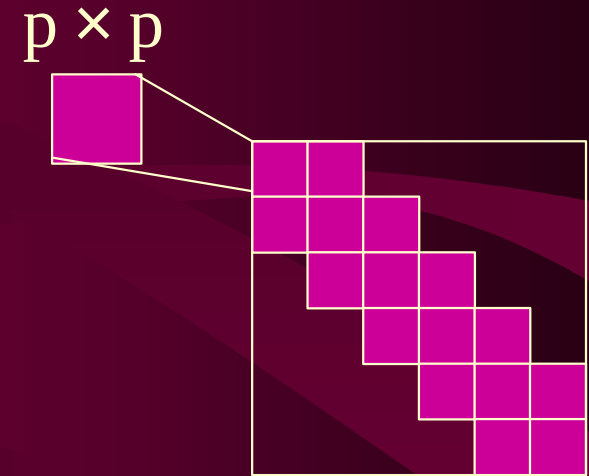


縮重度分の固有ベクトルが正しく求まらない可能性あり

ブロック化 Lanczos 法

- 原理 (Cullum and Donath, 1974)

- $p \times p$ の小行列を1つの要素と見て Lanczos 法を実行
- 出力はブロック三重対角行列



- 長所

- 縮重度が p までの縮重固有値・固有ベクトルが正しく計算可能 (Underwood, 1975)
- 計算が行列乗算を用いて行われるため, キャッシュマシンで高い性能を出すことが可能

ブロック化 Lanczos 法のアルゴリズム

42

列が正規直交ベクトルである初期ブロックベクトル V_1 を設定

$$\beta_0 = 0, \quad V_0 = 0$$

$$V_0 = \varnothing$$

DO J = 1, M

$$V_J = [V_{J-1} \mid V_J]$$

$$U = AV_J - V_{J-1} \beta_{J-1}$$

$$\alpha_J = V_J^t U$$

$$U := U - V_J \alpha_J$$

$$V_{J+1} \beta_J = U$$

END DO

英太太文字: 幅pのブロックベクトル
ギリシャ太文字: $p \times p$ 行列

Aの乗算による部分空間拡大
直交化

UのQR分解

$$T_M = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

$T_M s = \theta s$ を解いて θ, s を求める。

$\theta, x = V_M s$ をそれぞれAの固有値, 固有ベクトルとして採用

Thick Restart Lanczos 法 (1)

- 基本的なアイデア
 - Lanczos 法のステップ数に上限値 m を設ける。
 - 第 m ステップでは m 本の Ritz ベクトルを計算し, そのうち k 本 ($k < m$) を選んで初期ベクトルとし, Lanczos 法をリスタートする。
- メリット
 - リスタートを行っても, k 本分のベクトルの情報が保存される。
 - 基底ベクトルの本数の上限が m に固定される。
 - メモリ所要量の削減
 - 直交化の演算量削減

- 実現方法

- m-step Lanczos 分解 $AV_m = V_m T_m + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^t$ が入力
- T_m の k 本の固有ベクトルを並べた行列を Y とすると,

$$\begin{aligned} AV_m Y &= V_m T_m Y + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^t Y \\ AV_m Y &= V_m Y T_k^+ + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^t Y \\ AV_k^+ &= V_k^+ T_k^+ + \beta_m \mathbf{v}_{k+1}^+ \mathbf{s}^t \quad \cdots (*) \end{aligned}$$

ただし $V_m^+ = V_m Y$, $\mathbf{s} = Y^t \mathbf{e}_m$,

T_m^+ : Y に対応する固有値を並べた $k \times k$ の対角行列

- (*) 式は k -step Lanczos 分解と類似の形を持つ。
(右辺第2項に現れるベクトルが $\mathbf{e}_m^t$ でない点が異なる。)
- (*) 式を起点とし, Lanczos 法と同様に $\mathbf{v}_{k+2}, \mathbf{v}_{k+3}, \dots$ を計算していくことで, k 本のベクトルを保存してリスタートすることが可能

Thick Restart Lanczos 法 (3)

- $\mathbf{v}_{k+2}$ の計算

- $A\mathbf{v}_{k+1}$ を $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k+1}$ に対して直交化
- 以下, 右肩の $+$ を省いて書くと

$$\begin{aligned}\beta_{k+1} \mathbf{v}_{k+2} &= (\mathbf{I} - \mathbf{V}_{k+1} \mathbf{V}_{k+1}^t) A \mathbf{v}_{k+1} \\ &= (\mathbf{I} - \mathbf{v}_{k+1} \mathbf{v}_{k+1}^t - \mathbf{V}_k \mathbf{V}_k^t) A \mathbf{v}_{k+1} \\ &= (\mathbf{I} - \mathbf{v}_{k+1} \mathbf{v}_{k+1}^t) A \mathbf{v}_{k+1} - \mathbf{V}_k \beta_m \mathbf{s}\end{aligned}$$

- ここで, 前ページの(*) 式と基底の直交性より

$$\mathbf{V}_k A \mathbf{v}_{k+1} = \beta_m \mathbf{s}$$

- であることを用いた。

Thick Restart Lanczos 法 (4)

- $\mathbf{v}_{k+i+1}$ ($i = 2$) の計算
 - $A\mathbf{v}_{k+i}$ を $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k+i}$ に対して直交化

$$\begin{aligned}
 \beta_{k+i} \mathbf{v}_{k+i+1} &= (I - V_{k+i} V_{k+i}^t) A\mathbf{v}_{k+i} \\
 &= (I - \mathbf{v}_{k+i-1} \mathbf{v}_{k+i-1}^t - \mathbf{v}_{k+i} \mathbf{v}_{k+i}^t - V_{k+i-2} V_{k+i-2}^t) A\mathbf{v}_{k+i} \\
 &= (I - \mathbf{v}_{k+i-1} \mathbf{v}_{k+i-1}^t - \mathbf{v}_{k+i} \mathbf{v}_{k+i}^t) A\mathbf{v}_{k+i} - V_{k+i-2} (AV_{k+i-2})^t \mathbf{v}_{k+i} \\
 &= (I - \mathbf{v}_{k+i-1} \mathbf{v}_{k+i-1}^t - \mathbf{v}_{k+i} \mathbf{v}_{k+i}^t) A\mathbf{v}_{k+i}
 \end{aligned}$$

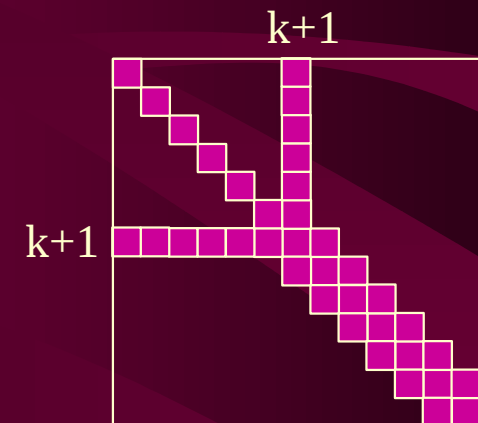
- ここで, AV_{k+i-2} は $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k+i-1}$ の線形結合であり, $\mathbf{v}_{k+i}$ に直交することを用いた。
- 上式より, $A\mathbf{v}_{k+i}$ は $\mathbf{v}_{k+i-1}, \mathbf{v}_{k+i}$ のみに対して直交化すればよい。
- したがって, リスタート後の各ステップの演算量は, 通常の Lanczos 法の演算量と等しい。

Thick Restart Lanczos 法 (5)

- Thick Restart Lanczos 法により得られる分解
 - $j = k+1$ のとき, 次の式が成り立つ。

$$AV_j = V_j T_j + \beta_j \mathbf{v}_{j+1} \mathbf{e}_j^t$$

- ただし, T_j は右図の形の $j \times j$ 行列である。



- 再リスタート

- 上式は T_j の非零構造を除いては k -step Lanczos 分解と同じ。
- 第 m ステップにおいて T_m の固有値・固有ベクトルを求めることにより, 再び (*) の形の式を構成でき, **再リスタート** が可能

性能評価 () ~ Lanczos法

- 評価環境

EP8000 /690Turbo

(Power4 1.3GHz)

- 評価例題

Matrix Market →

(Harwell-Boeing)

- 評価対象の手法

Lanczos vs. **TR-Lanczos**

収束に必要なベクトル本数

演算量・計算時間

収束状況(*bcsstk21*,*bcsstk39*)

Matrix	<i>n</i>	<i>nz</i>	<i>Gap</i>
<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	0.13
<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	0.34
<i>crystk02</i>	13965	491274	0.99

$$\text{密集度 } \mathbf{Gap} = (l_1 - l_{100}) / l_1$$

性能評価 () ~ Lanczos法

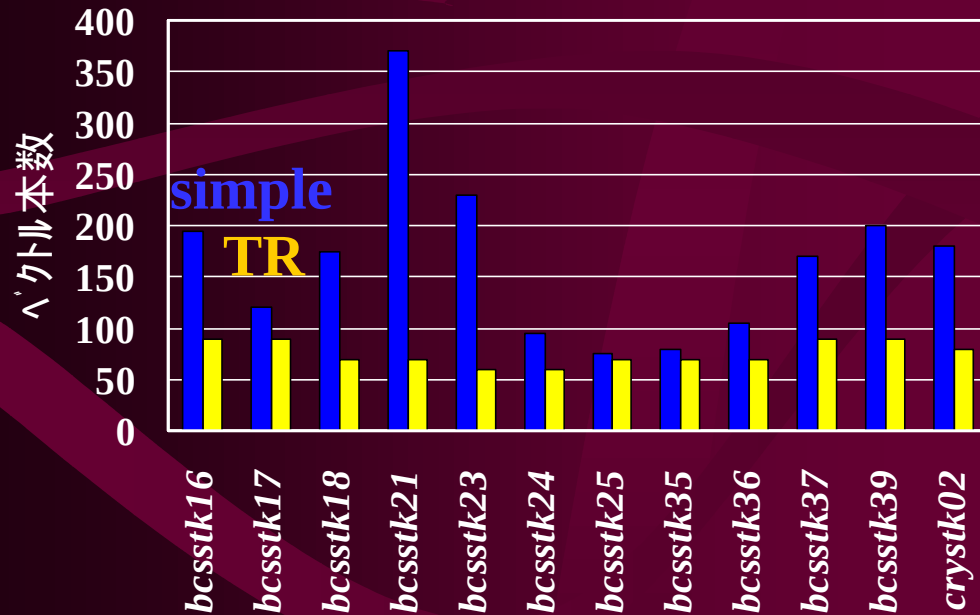
- 収束判定基準
- 所望固有値個数
- 結果

$$\|Ax - x\| / \|x\| < 10^{-10}$$

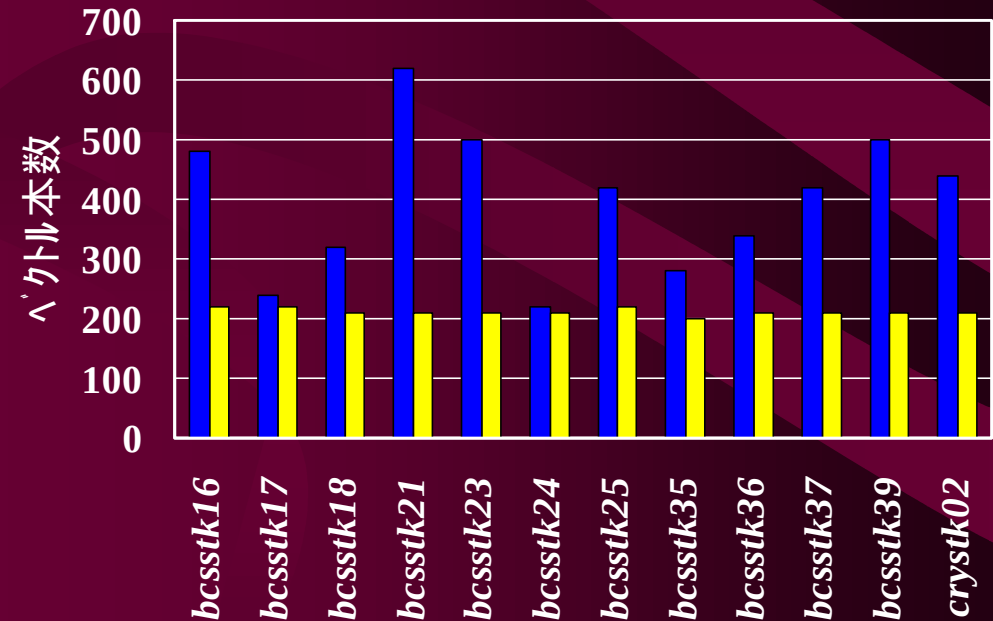
NE=30,100

TR:所要ベクトル本数少ない。
密集に対しても安定。

収束に必要なベクトル本数の比較



NE=30

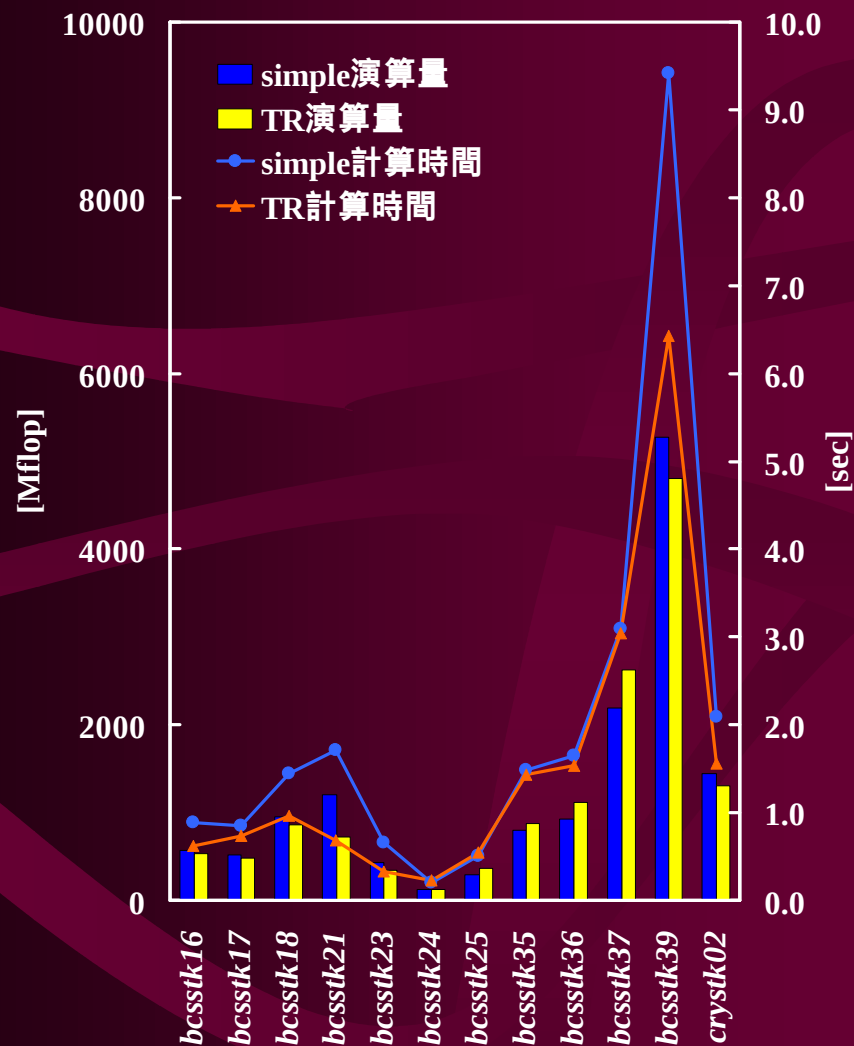


NE=100

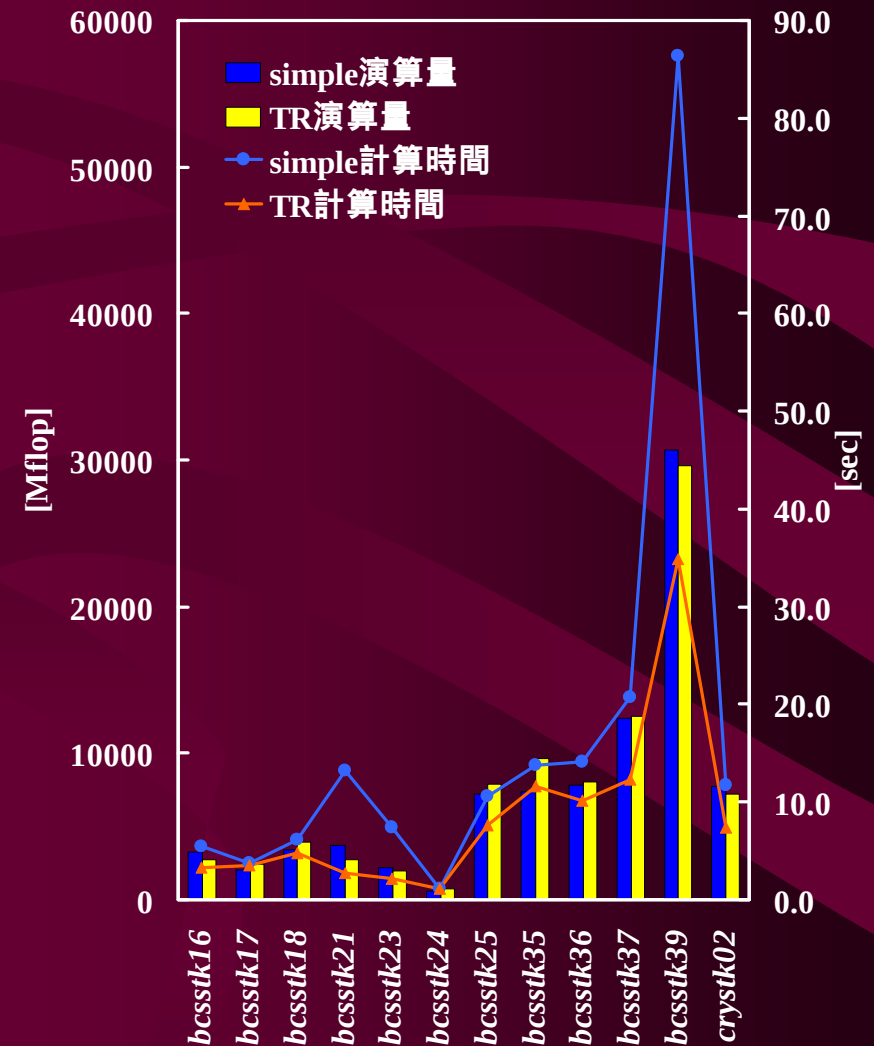
性能評価 () ~ Lanczos法

50

演算量は同等だが, 計算時間はTRが高速(**EP8000**) キャッシュマシン向き



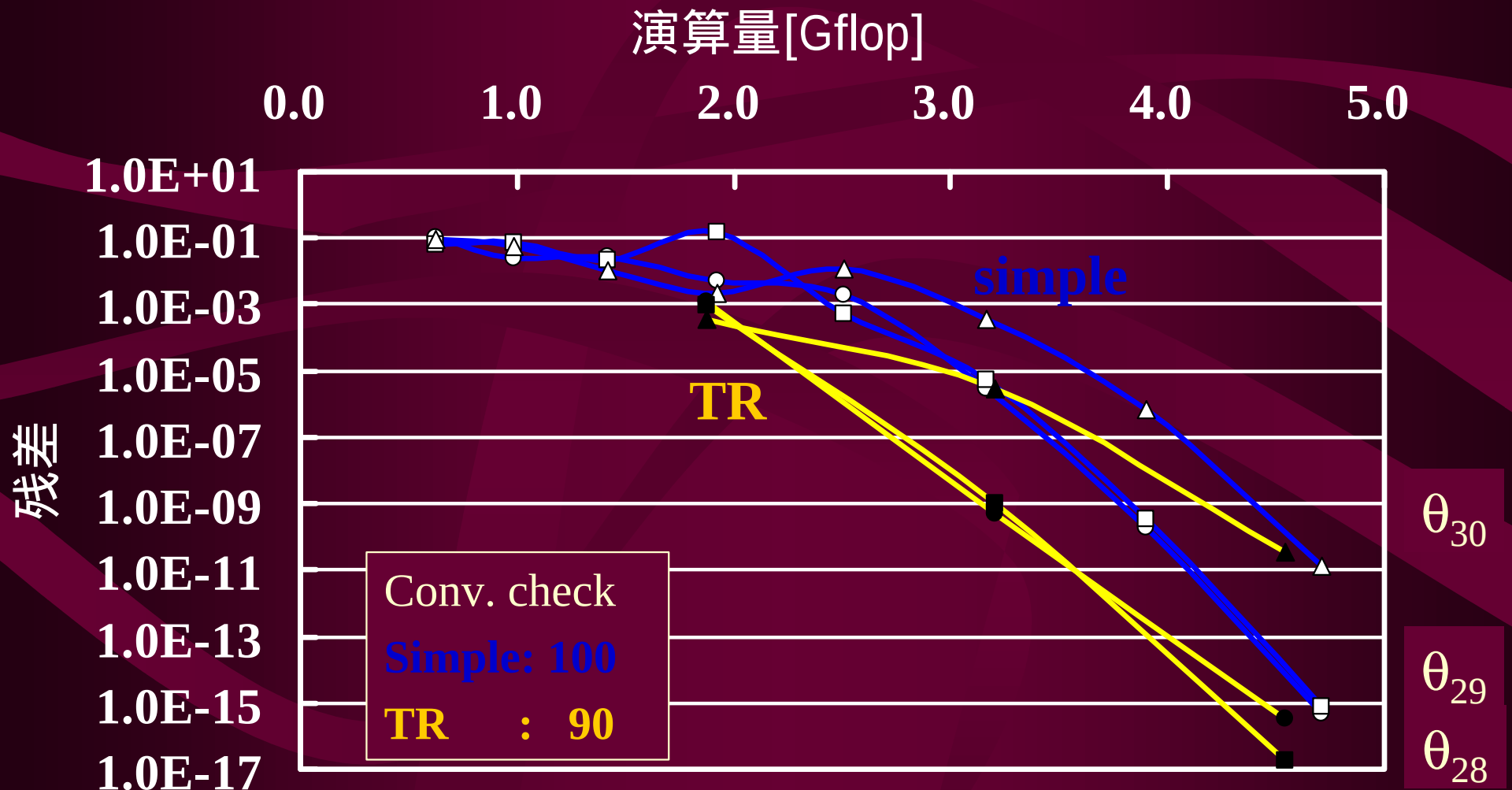
NE=30



NE=100

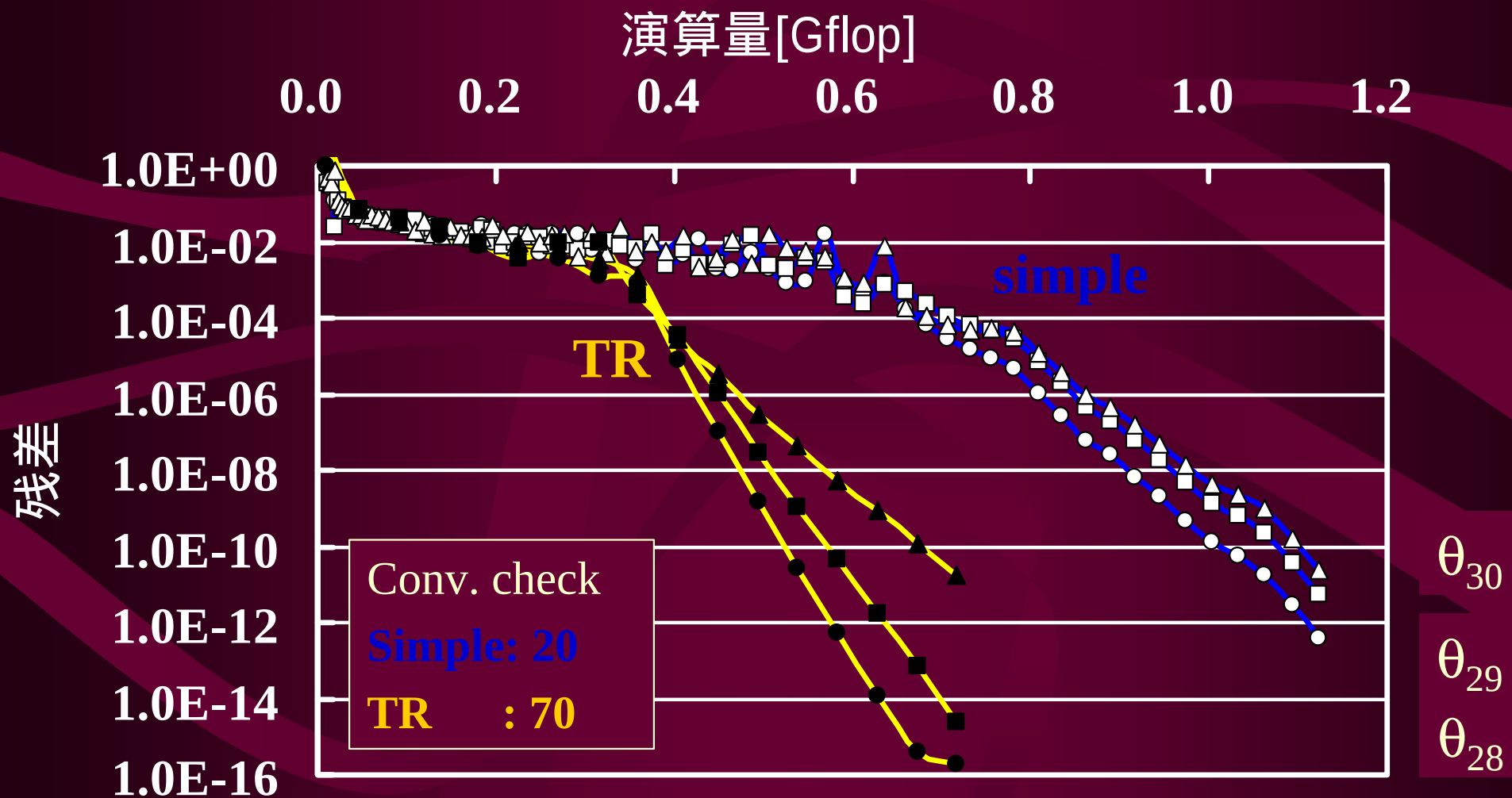
性能評価 () ~ Lanczos法 収束性(1)

- 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 () ~ Lanczos法 収束性(2)

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



5. Jacobi-Davidson 法

- Jacobi-Davidson 法を考える動機
- Jacobi-Davidson 法の原理
- アルゴリズム
- リスタート
- デフレーション
- 性能評価

Jacobi-Davidson 法を考える動機

- Lanczos 法 / Arnoldi 法で解きにくい問題
 - スペクトルの端にある固有値で、かつ他の固有値からの分離が良い場合は解きやすい。
 - それ以外の場合、効率的に計算を行うには、スペクトル変換などの技法が必要
 - しかし、 $(A - \sigma M)^{-1}$ による乗算は、コストが掛かる場合が多い。



スペクトル変換を用いずに、内部固有値 / 分離の悪い固有値を効率的に求められる解法が必要

Jacobi-Davidson 法の原理 (1)

- 修正ベクトルと部分空間の拡張
 - 部分空間 S_m と正規直交基底 $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ とが与えられているとする。 $V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$ とし, S_m に関する A の Ritz 値, Ritz ベクトルの組を $(\theta_j, \mathbf{u}_j = V_m \mathbf{s}_j)$ ($j = 1, \dots, m$) とする。
 - いま, 求めたい固有値 λ に対する近似固有ベクトルとして $\mathbf{u}_j$ を選び, $\mathbf{u}_j$ を正しい固有ベクトルにするための修正ベクトルを $\mathbf{t}$ と書くと,

$$\begin{aligned} A(\mathbf{u}_j + \mathbf{t}) &= \lambda(\mathbf{u}_j + \mathbf{t}) \\ \text{すなわち, } (A - \lambda I)\mathbf{t} &= -(A - \lambda I)\mathbf{u}_j \end{aligned}$$

- Jacobi-Davidson 法では, ベクトル $\mathbf{u}_j$ の直交補空間の中で $\mathbf{t}$ を求め, これによって S_m を拡張する。

Jacobi-Davidson 法の原理 (2)

- 修正方程式

- $(A - \lambda I)$ の作用を $\mathbf{u}_j$ に制限した演算子は

$$(I - \mathbf{u}_j \mathbf{u}_j^t)(A - \lambda I)(I - \mathbf{u}_j \mathbf{u}_j^t)$$

だから, $\mathbf{t}$ を求めるための方程式は

$$(I - \mathbf{u}_j \mathbf{u}_j^t)(A - \lambda I)(I - \mathbf{u}_j \mathbf{u}_j^t) \mathbf{t} = -(A - \lambda I) \mathbf{u}_j$$

となる。さらに, λ は未知だから θ_j で置き換えると,

$$(I - \mathbf{u}_j \mathbf{u}_j^t)(A - \theta_j I)(I - \mathbf{u}_j \mathbf{u}_j^t) \mathbf{t} = -(A - \theta_j I) \mathbf{u}_j$$

これを修正方程式と呼ぶ。

- JD法では, $\mathbf{t}$ を求めた後, これを $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ に対して正規直交化して $\mathbf{v}_{m+1}$ を求め, 部分空間を拡大していく。

Jacobi-Davidson 法の原理 (3)

- 修正方程式の解法

- 修正方程式の係数行列は u_j への制限のため特異であるが、右辺も u_j に属する。したがって方程式は無矛盾
- 初期値 $t_0 = 0$ から出発し、GMRES法、CGS法などの Krylov 部分空間法を用いて解けば、自動的に u_j に属する解が得られる。
- 経験上、厳密解を求める必要はなく、GMRES法の5ステップ分程度の精度で十分
- 前処理を行う場合は前処理行列 K も部分空間 u_j に制限する必要がある。

初期ベクトル $\mathbf{t} = \mathbf{v}_0$ を設定

DO $m = 1, 2, \dots$

DO $i = 1, 2, \dots, m-1$

$\mathbf{t} := \mathbf{t} - (\mathbf{t}^t \mathbf{v}_i) \mathbf{v}_i$] Modified Gram-Schmidt 法

END DO

$\mathbf{v}_m = \mathbf{t} / \|\mathbf{t}\|_2, \quad \mathbf{v}_m^A = A\mathbf{v}_m$

DO $i = 1, 2, \dots, m$

$M_{i,m} = \mathbf{v}_i^t \mathbf{v}_m^A$

END DO

$m \times m$ 行列 M の最大固有値 θ , 固有ベクトル $\mathbf{s}$ を求める。

$\mathbf{u} = V\mathbf{s} \quad (V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m])$

$\mathbf{u}^A = V^A \mathbf{s} \quad (V^A = A[\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m])$

$\mathbf{r} = \mathbf{u}^A - \theta \mathbf{u}$

$\|\mathbf{r}\|_2 \leq \varepsilon$ なら $\lambda = \theta$, $\mathbf{x} = \mathbf{u}$ として停止

$(I - \mathbf{u}\mathbf{u}^t)(A - \theta I)(I - \mathbf{u}\mathbf{u}^t)\mathbf{t} = -\mathbf{r}$ を解いて $\mathbf{t}$ を求める。

END DO

リスタート

- 必要性
 - 1ステップ当たりのメモリ所要量・演算量を一定値に抑えるため、JD法でもリスタートを行う。
- 実現方法
 - JD法では、部分空間 S_m が Krylov 部分空間の構造を持つ必要がないため、リスタートは容易
 - Ritz ベクトルのうち、求めたい固有値に近い Ritz 値を持つベクトルを複数本選び、部分空間の次元を削減して計算を続行すればよい。

- 必要性

- 複数の固有値・固有ベクトルを求める場合, 収束した固有ベクトルを基底から取り除いてリスタートすると, 計算が効率化できる。
- この操作をデフレーションと呼ぶ。

- 実現方法

- 収束した固有ベクトル $\mathbf{x}_1, \dots, \mathbf{x}_l$ を基底から取り除く。
- アルゴリズムにおいて, $A - \theta I$ を

$$(I - X_l X_l^t)(A - \theta I)(I - X_l X_l^t)$$

ただし $X_l = [\mathbf{x}_1, \dots, \mathbf{x}_l]$

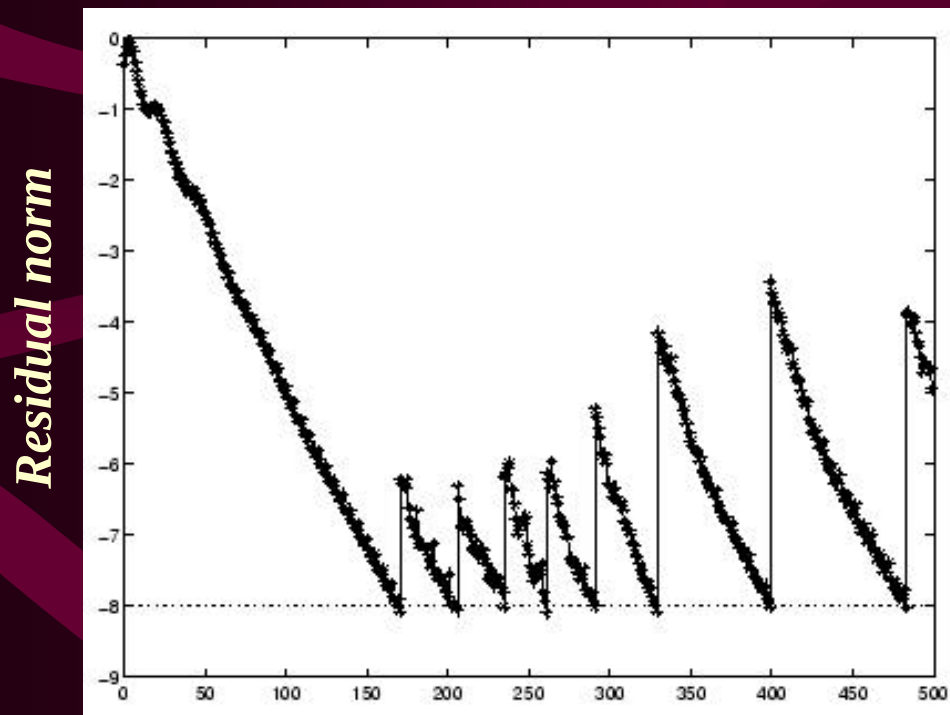
で置き換える。

- これにより, 計算はすべて X_l の中で行える。

性能評価 ~ JD 法 対称・収束特性

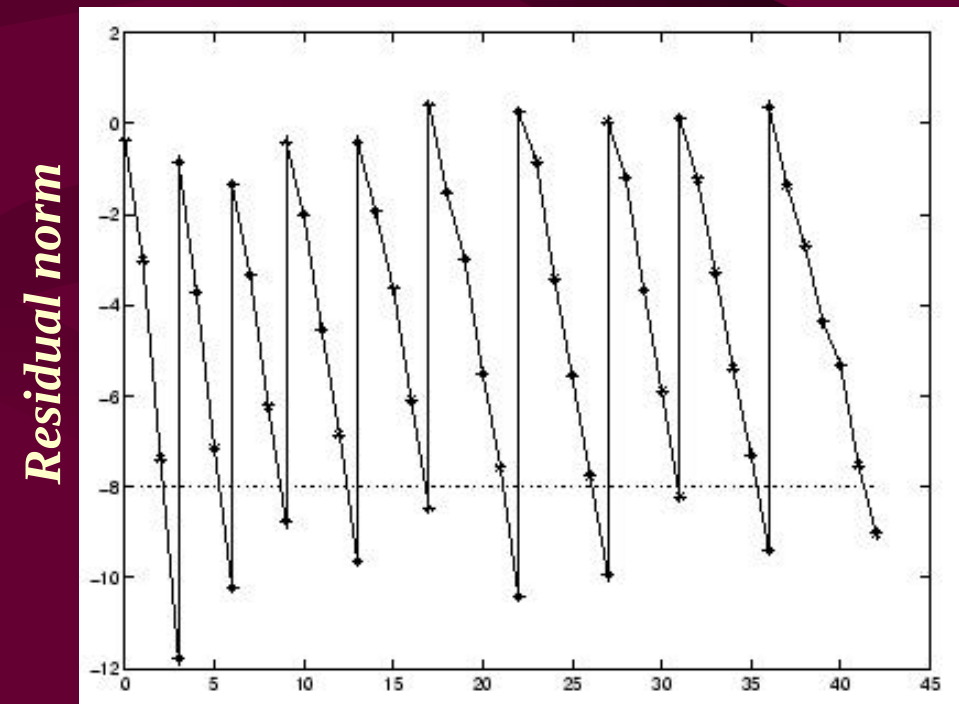
- テスト行列 $N=1000, \quad A_{ii}=i, \quad A_{i-1,i}=0.5, \quad A_{1000,1}=0.5$
- 収束判定 $\|Ax - x\| < 10^{-8}$

Arnoldi



Iteration number

JD with GMRES(5)



Iteration number

cf) Sleijpen, Jacobi-Davidson algorithms for various eigenproblems(1999),p.15 Fig.4-1

参考文献(1/5)

疎行列固有値計算全般

- 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.
- F. Chatlin: “Valeurs Propres de Matrices”, Masson, Paris, 1988. (W. Ledermann “Eigenvalues of Matrices”, John-Wiley and Sons, Chichester, 1993.)
- J. W. Demmel: “Numerical Linear Algebra”, SIAM, Philadelphia, 1996.
- G. H. Golub and C. F. van Loan: “Matrix Computations”, 3rd edition, The Johns Hopkins University Press, 1989.
- B. N. Parlett: “The Symmetric Eigenvalue Problem”, Prentice-Hall, Englewood Cliffs, 1980.
- Y. Saad: “Numerical Methods for Large Eigenvalue Problems”, Halsted Press, New York, 1992.
- J. H. Wilkinson: “The Algebraic Eigenvalue Problem”, Clarendon Press, Oxford, 1965.

参考文献(2/5)

Lanczos法(1)

- J. Cullum and W. E. Donath: “A Block Lanczos Algorithm for Computing the q Algebraically Largest Eigenvalues and a Corresponding Eigenspace of Large Sparse Real Symmetric Matrices”, Proc. of the 1974 IEEE Conference on Decision and Control, Phoenix, Arizona, pp. 505-509 (1974).
- J. Cullum and R. A. Willoughby: “Lanczos Algorithms for Large Symmetric Eigenvalue Computations”, Volume 1, Theory, Birkhauser, Boston, 1985.
- R. G. Grimes, J. G. Lewis and H. D. Simon: “A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems”, SIAM Journal on Matrix Analysis and Applications, Vol. 15, No. 1, pp. 228-272 (1994).
- C. Lanczos: “An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators”, J. Research of the National Bureau of Standards, Vol. 45, No. 4, pp. 255-282 (1950).
- 森正武, 杉原正顕, 室田一雄: “線形計算”, 岩波書店, 1994.

参考文献(3/5)

Lanczos法(2)

- B. N. Parlett and D. Scott: “The Lanczos Algorithm with Selective Orthogonalization”, Mathematics of Computation, Vol. 33, pp. 217-238 (1979).
- H. D. Simon: “The Lanczos Algorithm with Partial Reorthogonalization”, Mathematics of Computation, Vol. 42, pp. 115-136 (1984).
- R. Underwood: “An Iterative Block Lanczos Method for the Solution of Large Sparse Symmetric Eigenproblems”, Report STAN-CS-75-495, Department of Computer Science, Stanford University, Stanford, California (1975).
- K. Wu and H. Simon: “Thick-Restart Lanczos Method for Large Symmetric Eigenvalue Problems”, SIAM Journal on Matrix Analysis and Applications, Vol. 22, No. 2, pp. 602-616 (2000).

参考文献(4/5)

Arnoldi法

- W. E. Arnoldi: “The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem”, Quart. J. Applied Mathematics, Vol. 9, pp. 17-29 (1951).
- R. B. Lehoucq and D. C. Sorensen: “Deflation Techniques for an Implicitly Restarted Arnoldi Iteration”, SIAM Journal on Matrix Analysis and Applications, Vol. 17, No. 4, pp. 789-821 (1996).
- R. B. Lehoucq, K.J. Maschhoff.: “implementation of an implicitly restarted block Arnoldi method”, Preprint MCS-P649-0297, Argonne National Laboratory, IL, 1997.
- R. B. Lehoucq, D. C. Sorensen and C. Yang: “ARPACK User’s Guide”, SIAM, Philadelphia, 1998.
- D. C. Sorensen: “Implicit Application of Polynomial Filters in a k-Step Arnoldi Method”, SIAM Journal on Matrix Analysis and Applications, Vol. 13, No. 1, pp. 357-385 (1992).

参考文献(5/5)

Jacobi-Davidson法

- D. R. Fokkema, G. L. G. Sleijpen and H. A. van der Vorst: “Jacobi-Davidson Style QR and QZ Algorithms for the Reduction of Matrix Pencils”, SIAM Journal on Scientific Computing, Vol. 20, pp. 94-125 (1999).
- M. Genseberger, G. L. G. Sleijpen: “Alternative correction equations in the Jacobi-Davidson method”, Preprint No. 1073, 1999.
- M. Nool and A. van der Ploeg: “A Parallel Jacobi-Davidson-Type Method for Solving Large Generalized Eigenvalue Problems in Magnetohydrodynamics”, SIAM Journal on Scientific Computing, Vol. 22, No. 1, pp. 95-112 (2000).
- Y. Saad and M. H. Schultz: “GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems”, SIAM Journal on Scientific and Statistical Computing, Vol. 7, pp. 856-869 (1986).
- G. L. G. Sleijpen and H. A. van der Vorst: “A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems”, SIAM Journal on Matrix Analysis and Applications, Vol. 17, pp. 401-425 (1996).
- G. L. G. Sleijpen, H. A. Van der Vorst, Z. Bai : “Jacobi-Davidson algorithms for various Eigenproblems –A working document–”, Preprint nr.1114 ,1999.