

スツルム・ランチョス法

村田 健郎 (日立中研)

梅谷 征雄 (全上)

1 はじめに:

最近, ランチョス法系統の二つの論文:

Parlett・Scott の *hanczos with Selective Orth.* (1979, Math. Comp.)⁽²⁾

Ericsson・Ruhe の *Spectral Shift hanczos* (1980 Math. Comp)⁽⁴⁾

が相次いであらわれた.* これらはそのまゝでは汎用性に問題があり、メーカーのプログラムプロダクトとするには尚多くの検討, 改造を要すると推測されるが、両者を統合してやれば、汎用性は免も角として、かなり強力なものとなし得るのではないかと思われた。汎用性に関しては、Spectral Shift の戦略に問題ありと感じられるので、'免も角' と言ったわけである。

ところで、前報: スツルム・同時逆反復法のプログラム, GINVR⁽¹⁾Y において開発したグループ分け手法は、高度の汎用性があるから、GINVR⁽¹⁾Y のワケ組みを基本にして、GINVR⁽¹⁾Y の中の各グループ毎の (同時または単独) 逆反復のところを、

Parlett・Scott 流の選択的直交化つきランチョスにおきかえたものを作り、それと、GINVR⁽¹⁾Y そのものとの汎用性と性能比較を行って見たいと考えた。

* Parlett が最近書いた本⁽³⁾も併読されたい。

2 準備 (2, 3より)

[exact arith. による場合の主な定理]

真の固有値を λ_i , ランチョスを j ステップで切って T_j を作り、その固有値を $\theta_i^{(j)}$, $i=1, \dots, j$ (リッツ値) とする。

(j) をしばしば省略して単に θ_i と書く。 T_j を固有値解析して $T_j S_j = S_j \Theta_j$ としたとき、これと周知の関係:

$$A Q_j - Q_j T_j = \beta_j \rho_{j+1} e_j^* \quad (e_j^* = (0, \dots, \overset{j\text{番目}}{1}))$$

とを合せると、残差ベクトル $A y_i - y_i \theta_i$ (但し $y_i = Q_j s_i$) に関し、固有ベクトル s_i の j 要素を s_{ji} と書くとき、

$$A y_i - y_i \theta_i = A Q_j s_i - Q_j s_i \theta_i = \beta_j \rho_{j+1} e_j^* s_i,$$

これから、

$$[1] \quad \|A y_i - y_i \theta_i\| = \beta_j \|e_j^* s_i\| = \beta_j |s_{ji}|$$

(これは有用な関係である。プログラムの中でも使う)

次いで、残差ノルムと近似固有値についての関係から、

$$[2] \quad \text{各 } \theta_i \text{ に対し } \exists \epsilon_i; |\lambda_i - \theta_i| \leq \beta_j \|e_j^* s_i\| = \beta_j |s_{ji}|$$

これより積極的に (固有値の近接度と関係させて)

$$[3] \quad A z_i = \lambda_i z_i, \quad \gamma_i = \min_{k \neq i} |\lambda_k - \lambda_i| \quad \text{として、} \exists \epsilon_i;$$

$$|\lambda_i - \theta_i| \leq (\beta_j |s_{ji}|)^2 / \gamma_i$$

γ_i は計算できないが、代わりに $\delta_i = \min_{k \neq i} |(\theta_k \pm \beta_j |s_{ji}|) - \theta_i|$ がゼロでないならば、 γ_i のところに δ_i を代用できる。こうして、プログラムの中でも利用できるが、今回は使わなかった。

さてコンピュータ演算となると、上記のようには行かない。
まず直交関係 $Q_j^* Q_j = I$ が崩れる。 $Q_j^* Q_j$ の最小固有値の平方根を $\sigma_1(Q)$ とすると [2] は次の形をとる:

$$[4] \quad |\lambda_i - \theta_i| \leq \sqrt{2} (\beta_j |e_j^* s_i| + \|F_j\|) / \sigma_1(Q_j),$$

$$\text{ここに } F_j \text{ は、 } A Q_j - Q_j T_j = \beta_j q_{j+1} e_j^* - F_j \quad (\text{丸め誤差によるもの})$$

$\|F_j\|$ は、 $n * (\text{計算機eps}) * \|A\|$ のオーダー以下ゆえそう心配することはないが、 $\sigma_1(Q_j)$ が 1 から離れて殆んどゼロになることがある。これが困る。次の Paige の定理が、選択的直交化のキランチョスの基礎となっている:

$$[5-1] \quad y_i^* y_k = [g_{ii}(s_{jk}/s_{ji}) - g_{kk}(s_{ji}/s_{jk}) + f_{ik}] / (\theta_i - \theta_k)$$

$$[5-2] \quad y_i^* q_{j+1} = g_{ii} / \beta_{ji} = g_{ii} / (\beta_{ji} s_{ji})$$

ここに G, F は丸め誤差行列で、 $\|G\| \doteq \|F\| \doteq \sqrt{n} * (\text{計算機eps}) \|T_j\|$.

[5-1] : 収束していない リッツベクトル y_i, y_k はよく直交しており、一方、収束していない y_k には、よく収束している y_i の成分を多く含むと主張。

[5-2] : q_{j+1} には、よく収束している y_i の成分を多く含むと主張。

そこで、 q_{j+1} から、よく収束している y_i の成分をぬきとるという操作を各 j についてやって行けば、それで十分になるのではないか?。これが Parlett 等の考え方の大筋である*。今回この線に沿ってプログラムを作成した。それを次に示す。

* (2) には、他にいくつかの戦略が併記されているが、今回は (3) にのっているものに大筋を従った。

3 テストに使用のプログラムの体系と LANSO

{ $Ax = \lambda x$ の $0 < \lambda_i < a1$ なる λ_i と v_i ($i=1, \dots, nc$) を求める }
 $eps, eps0, epsl, epsq$ を与える. (例 $10^{-9}, 0.2 \times 10^{-9}, 10^{-6}, 10^3$)

SSTRM 使用の二分法 BISECT により, nc と粗いグループ分け情報 $nc[i], xu[i], xl[i], i=1, \dots, nc$ を得る.

合理的なグループ分けを行う. SSTRM を用い使用グループ番号 ig , 固有値個数 $ln[ig]$, グループ下限 $xug[ig]$, 上限 $xlg[ig]$, $igmax$ を決定

$k=1$

do $ig = 1, igmax$

{ ここまでは GINVR⁽⁹⁾ と同じ }

$$h = (xlg[ig] - xug[ig]) / 2.0$$

$$shift = xug[ig] + h \quad \{ ig \text{ 区間の中点} \}$$

①

$$B = A - shift * I \text{ を コレスキ } U^T D U$$

$$r = (DU)^{-1} e \quad \{ \text{ウイキンソンの初期ベクトル} \}$$

②

$$r = B^{-1} r$$

LANSO { LANczos with Selective Orth. }
(次頁)

400

CONTINUE

{ ここから先は GINVR と同じ }

LANSO で 収束しないもの: $difi > eps$ を RL 商シフトつき逆反復*で止めを刺す

$k = k + ln[ig]$ { k を次のグループの先頭に進める }

① : $shift$ 値が異常に固有値に近い, その他の理由でヒポット値がある程度以下になったときは, $shift$ 値を移動してやり直す. (GINVR と同じ)

② : 1回逆反復をほどこして $shift$ 値から遠い(グループの外の)固有値に対する固有ベクトル成分を予じめふるっておく. (結果が若干よい様子)

* Rayleigh 商を毎回作ってシフト値を更新しながら行なう逆反復, RL 逆反復と略称.

LANSSO の主要部

{こねよりランソス}

$q_1 = Y$; $u = B^{-1}Y$; $\alpha = q_1^T u$
 $r = u - \alpha * q_1$; $\beta = (r^T r)^{\frac{1}{2}}$
 $ta[1] = \alpha$; $tb[1] = \beta$; $norm = |\alpha| + |\beta|$
 $C = -1$
 $jmax = 16 + 8 * \ln[ig]$

③ do j = 2, jmax

$count = 0$
 $q = r / \beta$; $u = B^{-1}q$; $r = u - \beta * q_{j-1}$; $\alpha = q^T r$
 $r = r - \alpha * q$; $\beta = (r^T r)^{\frac{1}{2}}$; $q_j = q$
 $ta[j] = \alpha$; $tb[j] = \beta$; $norm = \max(norm, |\alpha| + 2|\beta|)$
 $C = C * (-1)$
if C = 1 then

④

三重対角行列 T_j の固有値の逆数 μ_{ji} ($i=1, \dots, j$) を求め、それを絶対値の小さい方から順にならべ、改めて μ_{ji} とする。

⑤ $-h - epsq < \mu_{ji} < h + epsq$ なる i の max を nqd とする。
 $difom = 0.0$

⑥ do i = 1, nqd {こねより selective Orth. }

T_j の固有ベクトル $S_i = (s_{1i}, \dots, s_{ji})^T$ を計算

$b_{ji} = |tb[j]| * |s_{ji}|$; $epsqn = epsq * norm$
if $b_{ji} < epsqn$ then

$y_i = Q_j S_i$; $\lambda_i = shift + \mu_{ji}$
 r から y_i 成分をぬきとる。
 $difo = \|A y_i - \lambda_i y_i\|$
 $epsc = epsl * norm$; $difom = \max(difo, difom)$
if $(difo < epsc) \wedge (-h \leq \mu_{ji} \leq h)$ count = count + 1

⑦

⑧ if $(count \geq \ln[ig]) \wedge ((j > L * (\ln[ig] + 2)) \vee (difo < eps0))$

go to 400

⑨ if count < $\ln[ig]$ ifusok = $\ln[ig] - count$

400 continue

- ③ : 例えば、グループ内の λ_i の個数 $\ln[ig]$ が 5 なら $j_{max} = 16 + 40 = 56$.
 まだ、これで本当によいかどうか確かでない。
- ④ : 毎回 T_j を固有値解析し、再直交化するのは馬鹿らしいからこうした。
- ⑤ : ngd は *number of good vectors* のつもり。
- ⑥ : ‘ $b_{ij} < \epsilon p_{qn}$ ’ であるようなベクトル s_i に対しては、いわゆるリツベクトル (もとの空間にひきのばされた $y_i = Q_j s_i$) を作り、 \mathbb{R} からその成分をぬきとる。ここが Parlett の LANSO のアイデアである。
- ⑦ : これで、また一本解ベクトルを捕捉したというわけである。
- ⑧ : この if 文は大変こみ入っているが、非常に重要である。(要するに、どこらあたりで LANSSO に見切りをつけて、最後の RL 商につつき逆反復にバトンを渡すのが適当か、というわけである。) 特に条件 :

$$j > L * (\ln[ig] + 2)$$

のためのパラメータ L は、GINVRY の経験からとりあえず決めているが、まだ自信はない。今のところ、 L は、 ϵ_{ps0} 値と関係させて、

$$\epsilon_{ps0} = 10^{-9} \sim 10^{-12} \text{ のときは } L = 4$$

$$\epsilon_{ps0} = 10^{-12} \sim 10^{-15} \text{ のときは } L = 5$$

と考えているが今回はすべて $L = 4$ のときのデータを載せた。

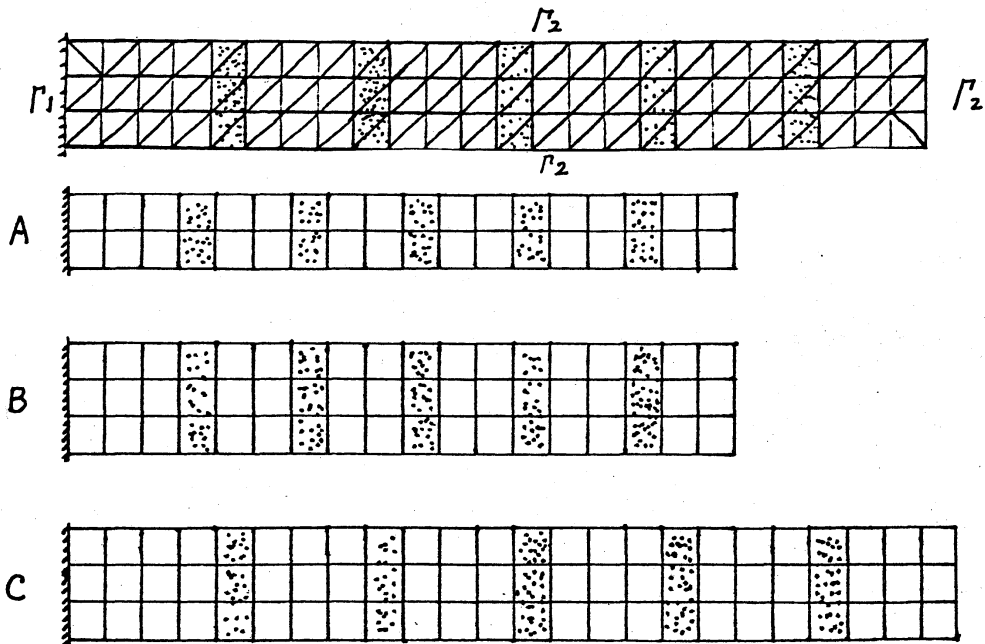
- ⑨ $count < \ln[ig]$ のことが、 ϵ_{psbi} の最適値使用のときは、A1, A2, A4, B1, B4, C1, C4 の諸ケースで、 $\epsilon_{ps0} = 0.2 * 10^{-14}$ までの範囲のテストケースで一度も起っていないが ϵ_{psbi} を故意に最適値の値から大きく外すと、盛に起る。

4 テスト問題とテストの現状

テスト問題として、とりあえず GINVRVY⁽¹⁾に使ったものの中から、A1, C1 (近接度の強い固有値をもつものの代表) 及び A4, B4, C4 (素直な固有値分布をしているものの代表) をえらんだ。

支配方程式は、 $\nabla \cdot (-k(x)\nabla u) = \lambda u \Rightarrow A u = \lambda u$

Γ_1 : 固定境界条件 $u=0$, Γ_2 : 自由境界条件 $\nabla u=0$



上図で、□部の k を 1.0 とし、■部の k を左から、(X=A, B, Cとして)

問題 X1	10^{-2}	10^{-5}	10^{-5}	10^{-8}	10^{-8}	例えは“Aの場で 物性定数 k を問題 1のようにえらん だとき、 <u>問題A1</u> と略称する。
問題 X2	10^{-1}	10^{-3}	10^{-3}	10^{-5}	10^{-5}	
問題 X3	10^{-1}	10^{-2}	10^{-2}	10^{-3}	10^{-3}	
問題 X4	0.31	10^{-1}	10^{-1}	10^{-2}	10^{-2}	

テストの現状：

epsbi値としては、0.4, 0.2, 0.1 の三ケースについて行なった。このテストに使用したRL逆反復のプログラムは、GINVRYに対してはこのテスト範囲で問題を起していなかったのであるが、LANSSOの後につなぐときには問題A1の次のケースで問題を起した。

$$\text{epsbi} = 0.4, \quad \text{eps} = 10^{-14}, 10^{-12}, 10^{-9}$$

$$\text{epsbi} = 0.2, \quad \text{eps} = 10^{-14}$$

$$\text{epsbi} = 0.1, \quad \text{eps} = 10^{-14}$$

問題 A1 の

0.633,974,597,5..., 0.633,974,599,3..., 0.633,974,604,3..., 0.633,974,604,9...

という4個の近接固有値の中の2個(一線)がRL逆反復過程で滞留して所望精度まで行くまえに反復5回打ちきりにかかるのである。テストの当時には、A1のような甚だしい近接固有値をもつ問題に対しては無理と誤認して、テストの主力をA4, B4, C4の系列に移してしまった。その後GINVRYでも、epsbiその他を故意に適値から大中に外すとき、同様の滞留現象が稀には起ることが判り、原因はRL逆反復に使用したコレスキにあることがつきとめられた。しかし、改良されたRL逆反復によってLANSSOの方を用テストしていないので、今回の報告は非常に不完全なものである。

上述の理由により、問題の素直な A4, B4, C4 の系列についての両者の比較を主として示そう。

$m = 160$ に換算した CPU カウントと USE カウントを示す。
(スツルム・同時逆反復法⁽¹⁾におけると同じ換算法をとる。)

問題 A4 $\epsilon_{ps} = 10^{-9}$, $al = 1.5$ ($nc = 23$)

最大変換本数 ↓

eps _{bi}	共通部		SUBSPACE (LAM=6)						LANSSO (L=4)								
	SSTRM	LU	反復	4% _m	6% _m	RL	TOTAL		反復	4% _m	6% _m	RL	TOTAL				
							CPU	USE					CPU	USE			
0.4	10	4	84	15	4	6	35	98	90	2	1	0	16	95	14		
0.2	15	6	88	12	3	6	39	103	92	2	1	2	25	101	14		
							平均 → 37	101								21	98

問題 B4 $\epsilon_{ps} = 10^{-9}$, $al = 1.5$ ($nc = 29$)

eps _{bi}	共通部		SUBSPACE (LAM=6)						LANSSO (L=4)								
	SSTRM	LU	反復	4% _m	6% _m	RL	TOTAL		反復	4% _m	6% _m	RL	TOTAL				
							CPU	USE					CPU	USE			
0.4	10	3	85	25	6	7	45	101	81	2	1	0	15	85	23		
0.2	17	6	107	17	4	5	43	122	122	3	1	1	27	130	17		
							44	112								21	113

問題 C4 $\epsilon_{ps} = 10^{-9}$, $al = 1.5$ ($nc = 29$)

eps _{bi}	共通部		SUBSPACE (LAM=6)						LANSSO (L=4)								
	SSTRM	LU	反復	4% _m	6% _m	RL	TOTAL		反復	4% _m	6% _m	RL	TOTAL				
							CPU	USE					CPU	USE			
0.4	10	4	104	25	6	6	45	120	94	2	1	1	17	100	18		
0.2	15	6	84	12	3	3	36	96	110	3	1	1	25	118	19		
							41	108								22	108

$\epsilon_{ps} = 10^{-9}$ のとき、

CPU カウントについては LANSSO 使用の方が SUBSPACE 使用のものにくらべて約半分、USE カウントについてはほぼ同じである。($m = 80$ で評価すると、LANSSO の方がもっと有利になる。) 次に $\epsilon_{ps} = 10^{-14}$ としたときどうなるかを示そう:

問題 A4 $\text{eps} = 10^{-14}$ $\text{al} = 1.5$ ($\text{nc} = 23$)

0.4	10	4	92	15	4	18	<u>47</u>	<u>118</u>	128	3	1	8	<u>25</u>	<u>141</u>	28		
0.2	15	6	100	12	3	9	<u>42</u>	<u>118</u>	146	4	1	8	<u>33</u>	<u>161</u>	23		
														45	116	29	156

問題 B4 $\text{eps} = 10^{-14}$ $\text{al} = 1.5$ ($\text{nc} = 29$)

0.4	10	3	98	29	7	13	<u>55</u>	<u>121</u>	143	4	1	21	<u>39</u>	<u>168</u>	23		
0.2	17	6	120	18	5	13	<u>54</u>	<u>144</u>	170	4	1	19	<u>46</u>	<u>196</u>	29		
														55	133	43	182

問題 C4 $\text{eps} = 10^{-14}$ $\text{al} = 1.5$ ($\text{nc} = 29$)

0.4	10	4	108	25	6	10	<u>49</u>	<u>128</u>	152	4	1	22	<u>40</u>	<u>129</u>	31		
0.2	15	6	120	18	5	14	<u>53</u>	<u>145</u>	170	4	1	8	<u>33</u>	<u>185</u>	28		
														51	137	37	182

$\text{eps} = 10^{-14}$ においても、CPU カウント については LANSSO 有利、USE カウント については LANSSO 不利となっているがこの程度のことなら、何れにしても大差はないということもできよう。

約言して、'LANSSO は、あまり甚だしい近接固有値をもたない素直な問題を、精度要求を極限まで追求すのでなく解きたいというときには大変有望である' と言えようか。

問題 A1, C1 の系列については先程述べたような歯切れの悪い事情があるが、C1 については LANSSO も正常に動作した

ので、それから先に比較しよう。

問題 C1 $\epsilon_{PS} = 10^{-9}$ $\alpha_l = 1.5$ ($n_c = 30$)
 $LQM = 5$ $L = 4$

epsbi	共通部		SUBSPACE					Total		LANSSO					Total		
	SSTRM	LU	反復			RL	CPU	USE				RL	CPU	USE			
0.4	9	4	60	12	3	2	27	69	138	3	1	5	21	193	17		
0.2	13	6	61	7	2	3	29	72	122	3	1	1	23	132	14		
0.5	16	7	71	8	2	3	34	83	153	4	1	2	29	163	32		
							30	75								23	196

問題 C1 $\epsilon_{PS} = 10^{-14}$ $\alpha_l = 1.5$ ($n_c = 30$)
 $LQM = 5$ $L = 4$

	9	4	95	25	6	7	45	112	156	4	1	22	39	183	29		
	13	6	97	15	4	4	38	111	174	4	1	8	31	189	32		
	16	7	91	11	3	8	42	109	183	5	1	25	53	216	32		
							42	111								41	196

以上 C1 においては $\epsilon_{PS} = 10^{-14}$ のとき CPU カウントは同じぐらい、USE カウントは LANSSO の方が 2 倍に及ぶ。

問題 A1 についてのデータには問題があるが、一応あげる。
 問題 A1 $\epsilon_{PS} = 10^{-9}$ $\alpha_l = 2.0$ ($n_c = 30$)

	14	5	72	11	3	1	31	81	129	3	1	14*	36*	149	13		
	20	7	76	10	3	0	37	86	117	3	1	0	30	125	14		
	24	9	88	7	2	1	41	100	119	3	1	0	26	129	16		
							36	89								34*	134

* 27 滞留

問題 A1 $\epsilon_{PS} = 10^{-14}$ $\alpha_l = 2.0$ ($n_c = 30$)

	14	5	90	15	3	8	42	107	161	4	1	31*	57*	206	2		
	20	7	97	13	3	1	41	108	163	4	1	35*	66*	205	3		
	24	9	102	10	3	5	48	118	175	4	1	28*	64*	163	11		
							44	111								55*	191

* 37 滞留

*印のものはRL逆反復が滞留して反復最大値6までで打ちきりになったものがあることを示す。A1, $\epsilon = 10^{-9}$, $\epsilon_{\text{sb}} = 0.4$ のとき2個, A1 $\epsilon = 10^{-14}$ のときはそれぞれ3個ある。RL逆反復のプログラムを改良すれば、それぞれのdifoの内容からみて()内の値で収束すると考えている。とすると、A1においては $\epsilon = 10^{-14}$ のときCPUカウントはLANSSOの方が若干多い程度でありUSEカウントはLANSSOが約2倍程度というイメージとなろう。

結局約言すれば、‘LANSSO使用のストルム・ランテヨス・RL逆反復法は、甚だしい密集固有値をもつ問題を計算機の精度の限界近くまでの精度で求めようとするときには問題であるが、そうでない問題領域で、ストルム^{同時}・逆反復・RL逆反復よりも有利な分野が、可なりありそうである’と言えようか、しかし、LANSSOは、すでに述べたこと以外にも、テスト段階でいろいろデリケートな振舞に出会っているので、汎用化のためには、これから尚多くの工数を必要としよう。

なほ、問題A1の場合、Parlett流の選択的直交化なしでは全く使えなかったこと、また30個の固有値を一度にやらせようとしたときには、LANSSOでもうまく行かなかったことを附言しておきたい。

文献

- (1) 村田, 後 'スツルム・同時逆反復法' 本講究録
- (2) Parlett・Scott 'Lanczos with Selective Orth' (1979, Math. Comp.)
- (3) Parlett 'The Symmetric Eigenvalue Problems' (1980 Prentice H.)
- (4) Ericcson・Ruhe 'Spectral Shift Lanczos' (1980 Math. Comp.)