

条件つき最小自乗問題における

代表的アルゴリズムの比較について

城西大理 西尾 美知子 (Michiko Nishio)

城西大理 中山 隆 (Takashi Nakayama)

原子力研 藤村 統一郎 (Touichirou Hujimura)

[1] はじめに

東京大学大型計算機センターで開発された最小自乗法プログラム S A L S にない機能を追加すること，および数値解法の改良が本研究の目的である．すなわち条件つき最小自乗法を追加することがそのひとつで，その為の各アルゴリズムの性能比較を行った．数値実験の結果，条件つき最小自乗法の 3 つのアルゴリズムのうちでは Lagrange の方法が最良となった．さらに第 2 の数値解法の改良として，従来の正規方程式法の欠点を補う意味で新しく採用した行方向 Q R 法は，予想通り，係数行列 A の次元 n が大きい時，計算時間および精度の両面に優れていることが実証された．

アルゴリズムの比較に用いた例題は m コのデータセット

$\{x, y\}$ を n コの 3 次スプライン基底で最小自乗 $f i t$ する問題である．この場合，係数行列 A は第 1 図のようなプロ

ックバンド行列となる。また，得られ
 る適合曲線は3点で極値をもつので，
 条件として各点における $f(x)$ の導関
 数を0に等しいとすれば，条件行列 C
 の形は第2図のようになる。

* * * * 0 0 0
 * * * * 0 0 0
 * * * * 0 0 0
 0 * * * * 0 0
 0 * * * * 0 0
 0 * * * * 0 0
 0 0 * * * * 0
 0 0 * * * * 0
 0 0 * * * * 0
 0 0 0 * * * *
 0 0 0 * * * *
 0 0 0 * * * *

第1図 行列 A

* * * * 0 0 0
 0 * * * * 0 0
 0 0 0 * * * *

第2図 行列 C

条件つき最小自乗法の問題はつぎのようになる。

$$\min \|Ax - b\|, \quad Cx = d$$

ただし $A : m \times n$, $C : t \times n$, $b : m \times 1$, $x : n \times 1$

$$t \leq n \leq m, \quad t < 10.$$

われわれの目的は前述したように

(1) 正規方程式法とQR法の比較 (QR法が有利という
予想のもとに)

(2) 条件つき最小自乗法の3つのアルゴリズムの比較
である。しかしその前に、条件のない ($Cx = d$ のない) 最小自乗問題を解く代表的な3つのアルゴリズムを、つぎに概観してみよう。

(2) 条件なしの最小自乗問題の解法

(1) 正規方程式法 (以下NE法と略)

NE法は従来からある方法で、連立方程式

$$A^T A x = A^T b$$

を解く (A^T は A の転置)。そのアルゴリズムはつぎの
ようになる。

a. $B = A^T A = L L^T$ とCholesky分解。

b. 連立方程式

$$L y = A^T b, \quad L^T x = y \quad \text{を解く。}$$

この方法の問題点として、つぎの様なことがあげられる。
 まず、 B の条件数は A の条件数の 2 乗になるので、連立方程式 $Bx = b'$ の解の精度は少し不良な A の場合、相当疑問である。また B の計算は A の転置行列と A の積であるから、そのときに情報の損失がある。さらに、 A が sparse 行列でも B が sparse とは限らない。

これらの理由で、伝統的な NE 法はいずれより良いアルゴリズムにとって代られるであろう。つぎに述べる QR 法などはその有力な候補の 1 つである。

(2) QR 法

a. $A = Q(R, 0)^T$ と QR 分解する。

ここに $Q : m \times m$ の直交行列、

$R : n \times n$ の上三角行列。

b. $Q^T b = (c, d)^T$, $c : n \times 1$,

$d : (m - n) \times 1$

c. 連立方程式 $Rx = c$ を解く。

A の QR 分解を得るための方法として, Gram-Schmidt法, Householder 法, Givens法等がある. 一般に sparse問題に直交変換を用いる場合, 途中でゼロでない要素が発生する欠点がある. これに対し行毎のGivens法は他の2つの方法と違って途中のふくれがない. 行列 A の上の部分 ($n \times n$) が上三角行列 R になるように, A の各行を処理してゆく. その様子は下の如くである.

$$\begin{vmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \end{vmatrix}$$

第 3 図

$$\begin{vmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{vmatrix}$$

第 4 図

Givens法は column oriented が普通であるが, row oriented も可能でこの方が良いこともある. 例えば A を $n \times 5$ の行列とし, $n > 4$ とする. そして A の最初の 3 行に Givens 変換を施して, 第 3 図のようになったとする. つぎに第 4 行を処理する. ここで, Givens変換をそれぞれ (1, 4) 行, (2, 4) 行, (3, 4) 行に順次施して第 4 図を得る. そ

してこれを最後の行に至るまで繰り返し適用する。この行毎のGivens法は、今回NE法の代りに新しく採用して、うまくいったと思われる方法である。

(3) PW法

PW法は、Peters - Wilkinson 法の略である。

a. 順列行列 P , Q を選んで

$PAQ = LU$ (LU 分解) とする。

ここに $L : m \times n$ の下台形行列,

$U : n \times n$ の上三角行列 である。

b. $Pb = d$ において

$\min \|Lz - d\|$ の最小自乗問題を解く。

c. つぎに $Uy = z$, $x = Qy$ より x を求める。

PW法は係数行列 L が単位下台形であるので、通常のNE法でも解は安定であるといわれている。 P と Q は、安定性と sparse性を考慮して選ぶ。 L の条件は良いが、 U には初めの行列 A の不良条件性がうつされる。したがって、 $Lz \sim d$ の最小自乗問題はNE法でうまくゆくと思われる。今回の実験的研究では、このPW法のテストは行っていない。

以上が条件なしの最小自乗法の3つの解法のあらましであるが、次節以降、条件つき最小自乗法の代表的な3つの解法の比較検討に移ろう。

[3] 条件つき最小自乗法の解法

条件つき最小自乗法の解法はいずれも条件なしの問題に変換して行うので前述した解法との組合せになる。

(1) Projection法

この型の方法の基本的考えかたは次の如くである。

まず条件式 $Cx = d$ を満足する点の部分空間のみを計算の対象とする。

いま、 y_1 を $Cx = d$ の特殊解とする。たとえば、

y_1 を $Cx = d$ の最小自乗解とする。

$$\text{すなわち, } y_1 = C^T (CC^T)^{-1} d$$

そうすると、 $Cx = d$ のすべての解は

$$y_1 + Z \bar{y}_2$$

で表わされる．ここに Z は C の null 空間の基底である．

この式を $Ax = b$ に代入して次式となる．

$$\min \| (AZ) \bar{y}_2 - (b - Ay_1) \|_2$$

上の最小自乗解は，もし AZ が rank $n - t$ を持てば一意であることに注意されたい．ここで Z の選びかたが問題となる． Z として直交基底をとるのは無難であるが，大型の問題では Z をなるべく小さい空間（行列 Z の要素にゼロ要素が多い）に表わした方がよい．以下は，ある程度上述のことを考慮した Projection 法のアルゴリズムである．

$$a. \ C^T = Q(R, 0)^T$$

と QR 分解する．

ここに $Q : n \times n$, $R : t \times t$

$$b. \ Z \text{ をつくる．すなわち } Q = \{Q1, Q2\},$$

$Q1 : n \times t$, $Q2 : n \times (n - t)$ と分けて，

$Z = Q2$ とする．

$$c. \ R^T \tilde{y} = d, \ R \hat{y} = \tilde{y},$$

$$y_1 \leftarrow C^T \hat{y}$$

$$d. \quad \bar{b} \leftarrow b - A y_1$$

e. $A Z$ をつくる.

f. $\min \| (A Z) \bar{y}_2 - \bar{b} \|$ を解く.

$$y_2 \leftarrow Z \bar{y}_2$$

$$g. \quad x \leftarrow y_1 + y_2$$

(2) 消去法

これは1種のProjection法であることに注意.

行列

$$\begin{pmatrix} C \\ A \end{pmatrix} = \begin{pmatrix} C1 & C2 \\ A1 & A2 \end{pmatrix}$$

と分解する. ここに行列 $C1$ は $t \times t$ の正則行列になるように列交換をするものとする. しかし実際のプログラムではこの列交換が案外難しい. また $C1$ が正則であっても $|C1|$ が0に近い場合は, 得られる解の精度はかなり悪い. これは消去法の欠点である.

C の分割に対応して $x = (x_1, x_2)$ とし,

この x_1 について条件式を解いて、さらにその式を

$\|Ax - b\|$ に代入して、つぎの条件なしの最小自乗法の式を得る.

$$\min \| \bar{A} x_2 - \bar{b} \|$$

$$\text{ここに } \bar{A} \leftarrow A_2 - A_1 C_1^{-1} C_2,$$

$$\bar{b} \leftarrow b - A_1 C_1^{-1} d$$

さて、つぎのようにおくと、 Z は C の零空間の基底、また

y_1 は $Cx = d$ の 1 つの解となる.

$$Z = \begin{pmatrix} -C_1^{-1}C_2 \\ I \end{pmatrix}, \quad y_1 = \begin{pmatrix} C_1^{-1}d \\ 0 \end{pmatrix}$$

このように、消去法はProjection法の特別な場合であることがわかる. 条件なしの最小自乗法の解法はProjection法と同様に、QR法と正規方程式法の両方で解かれる.

(3) Lagrange の乗数法

Lagrange法は

a. 条件式の数が多い

b. 行列 A は full column rank である

とき有利. そうでないときは前の2つの方法より一般的でないといわれる. われわれの例は上の条件を満たしていたので, Lagrange法の結果は良好であった.

まず, つぎの2次計画問題を考える.

$$\min \quad 1/2 \{ x^T (A^T A) x$$

$$- (A^T b)^T x \}, \quad Cx = d$$

上の式の解の存在するための必要十分条件は, つぎの式を満足させるベクトル λ の存在することである.

$$\begin{pmatrix} A^T A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} A^T b \\ d \end{pmatrix}$$

この連立方程式はブロック行列の形で次のように解ける.

$$1. \quad (A^T A) y = A^T b,$$

$$2. \quad [C (A^T A)^{-1} C^T] \lambda = C y - d$$

$$3. \quad x \leftarrow y - (A^T A)^{-1} C^T \lambda$$

しかし H e a t h は λ について解く必要はないことを示した。そのアルゴリズムはつぎの如くである。

$$a. \quad A = Q (R, 0)^T \text{ と分解する.}$$

$$\text{ここに } Q : m \times m, R : n \times n$$

$$b. \quad \text{連立方程式 } R y = c \text{ を解く } (c = Q^T b)$$

$$c. \quad t \text{ の連立方程式}$$

$$R^T K = C^T \text{ を解く.}$$

$$d. \quad K = \tilde{Q} L^T \text{ と QR 分解する.}$$

$$\text{ここに } \tilde{Q} : n \times t \text{ の行列で列は正規直交である.}$$

$$e. \quad r \leftarrow d - C y$$

$$f. \quad \text{連立方程式 } L s = r \text{ を解く.}$$

$$g. \quad v \leftarrow \tilde{Q} s$$

$$h. \quad \text{連立方程式 } R z = v \text{ を解く.}$$

$$i. \quad x \leftarrow y + z$$

前述したように、 A が full rank で、 t が小さければ、Lagrangeの方法は次元の大きい問題に対して非常によい。

ステップ a. では、勿論、 R の要素に 0 が増加しないように、初めに行列 A に列交換の処理をしておくといよい。しかし今回はその処理をしていない。

条件が同次のとき等、Lagrange法の別の解釈もある。

(4)

sparse行列の問題においてその計算時間およびstorageの大きさは、行列の 0 要素の多寡が関係してくる。したがって各アルゴリズムの比較にあたって表われる行列の形がいかに変わるかが問題になる。

Projection法において行列 Z 、 AZ はそれぞれどういう形であろうか。

まず、行列 A の形は第 1 図に示したようなブロックバンド行列であるから、 $A^T A$ の形 (7×7) は第 5 図の様に

なる。また、 A を QR 分解したときの R の形 (7×7) を第 6 図に示す。

* * * * 0 0 0

* * * * * 0 0

* * * * * * 0

* * * * * * *

0 * * * * * *

0 0 * * * * *

0 0 0 * * * *

* * * * 0 0 0

0 * * * * 0 0

0 0 * * * * 0

0 0 0 * * * *

0 0 0 0 * * *

0 0 0 0 0 * *

0 0 0 0 0 0 *

第 5 図 $A^T A$ の形

第 6 図 $A = QR$ の R の形

上の第 5 図と第 6 図を比較すればわかるように,

$B = A^T A$ と QR 分解した R とはよく似ている. すなわ

ち $A^T A$ は対称行列であるが, その上半分をとれば R に

なっている. P a r t e r によれば,

(L e m m a) $B_{IJ} = 0$ ならば, $R_{IJ} = 0$

または ある $k < \min \{i, j\}$ があって

$R_{ki} = 0$ かつ $R_{kj} = 0$.

つぎに， A の QR 分解の直交行列 Q の形を第 7 図に示す．

<pre> * * * * * * * * * * * * * * * 0 * * * * * 0 * * * * * 0 * * * * * 0 0 * * * * * 0 0 * * * * * 0 0 * * * * * 0 0 0 * * * * * 0 0 0 * * * * * 0 0 0 * * * * * </pre>	このように Q の形には一定の規則がない．同様に条件行列 Z の形にも一定の規則がない．まして AZ の形の規則性はない．しかし，今後の研究によってその構造が解明されるかもしれない．現在のところ，AZ に表われる非零要素に対してはそれを少なくするような処理をしていない．
--	--

第 7 図 直交行列 Q の形

[5] 結果

さて，ここで計算時間および精度について実験の結果をまとめてみよう．

(イ) 時間についてはつぎの表を参照されたい．

(所要時間の単位はm s e c)

(m, n)	P R O	E L I	L A G	* P R O	* E L I
(12, 7)	7	5	9	4	3
(12, 8)	8	5	11	6	4
(12, 9)	9	6	11	7	5
(12, 10)	10	7	12	9	6
(12, 11)	11	8	13	10	7
(12, 12)	13	8	13	12	8

備考 1) m, n : 行列の行, 列の数

2) P R O, E L I, L A G : それぞれProjection法,
消去法, Lagrangeの乗数法

3) * はQ R法の代りに正規方程式法を用いた。

第1表 各アルゴリズムの所要時間の比較 (その1)

mが12位の間は各方法ともあまり差はない。かえって正規方程式法を用いた方がわずかに早い位である。

しかし、次にみるようにmが大きくなってゆくと、たとえばm = 111のときQ R法が逆転して以後その差は開くばか

りとなる。すなわち

(m, n)	PRO	ELI	LAG	*PRO	*ELI
(111,10)	84	75	68	36	27
(111,20)	213	149	121	167	105
(111,30)	'374	"210	180	'392	"234
(111,40)	583	272	243	716	410
(111,50)	842	329	313	1142	646
(111,100)	2835	594	705	4777	2567

第2表 各アルゴリズムの所要時間の比較（その2）

‘や”の印のついた行はQR法が正規方程式法より勝り始める行である。

以後の結果をまとめて1表にする。この表から、所要時間がm, nの関数としてどのようなになっているか推測することが出来る。

(m, n)	P R O	E L I	L A G	* P R O	* E L I
(1101,10)	830	757	623	323	250
(1101,20)	2127	1539	1035	1578	996
(1101,30)	3741	2171	1454	3764	2206
(1101,40)	5863	2830	1881	6886	3887
(1101,50)	8463	3497	2311	10961	6064
(1101,100)	28525	6799	4591	45420	23913
(4401,10)	3320	3025	2471	1298	1014
(4401,20)	8521	6171	4083	6343	4022
(4401,30)	14968	8695	5705	15140	8932
(4401,40)	23445	11371	7331	27714	15765
(4401,50)	33829	14090	8969	44254	24626
(4401,100)	113976	27543	17338	183153	97529

第 3 表 各アルゴリズムの所要時間の比較 (その 3)

最後の行を見れば, 5つの方法の中でLagrange法が最もすぐれていることがわかる. そのとき, すぐ右のProjection法で正規方程式法を用いたのは, その10倍の時間を要している.

*印のついた正規方程式法の所要時間は n^2 のオーダー

であるのに対し、QR法のそれは n のオーダーであり、これは A が sparse のとき一般になりたつ。

(ロ) つぎに精度の比較にうつる。24 の case 全般についていえることは

- (1) 消去法による結果が悪い。それは選択された正則行列 $C1$ の条件が不良のときである。すなわち、上三角行列 $C1$ の対角要素が 10^{-3} 以下の場合、

得られた解は大きな誤差をもつ。

- (2) m が大きくなると全般的に精度が落ちる。しかし例外として $m = 4401$, $n = 100$ のとき5つのアルゴリズムとも解は一致した。

以上のことを示す3つの case を、第4表から第6表に掲げる。

***** C1 *****
 -0.478 0.005 0.0
 0.0 -0.001 0.0
 0.0 0.0 -0.430

シ"ヨウカ" ヲキ サイヨウ シ"シ"ヨウカイ (M = 1101, N = 20)

*** PROJECTION ***
 0.988 1.393 2.302 2.909 3.441 3.065 3.034 1.798 1.831 1.795
 2.447 2.917 3.494 3.946 4.257 4.076 3.729 2.757 1.162 0.754

*** ELIMINATION ***
 1.031 1.385 2.305 2.920 3.437 3.072 2.989 1.802 1.829 1.800
 2.444 2.919 3.493 3.947 4.256 4.077 3.729 2.757 1.162 0.755

*** LAGRANGE ***
 0.982 1.394 2.302 2.909 3.441 3.064 3.034 1.798 1.831 1.795
 2.447 2.917 3.494 3.946 4.257 4.076 3.729 2.757 1.162 0.754

*** PROJECTION (NORMAL) ***
 0.980 1.395 2.302 2.909 3.441 3.064 3.034 1.798 1.831 1.795
 2.447 2.917 3.494 3.946 4.257 4.076 3.729 2.757 1.163 0.753

*** ELIMINATION (NORMAL) ***
 2.158 1.125 2.487 2.709 3.530 2.945 3.551 0.728 3.131 0.287
 3.360 2.415 3.764 3.801 4.342 4.022 3.762 2.730 1.200 0.592

第5表 精度 (m = 1101, n = 20)


```

*****
**** ショウケン ツキ LS モンタ"イ ( ショウウ シ"カン ノ ヒカク ) T E S T メイン プログラム ****
*****
*
  INTEGER T
  INTEGER TIME(24,5),MM(24),NN(24),CH(3)
  REAL    XX(12),YY(12),Y(4401),DX(3),X1(100),W1(4401),W2(4401)
  REAL    A(4401,100),C(3,100),Q(4401,100),Q2(100,100)
  REAL    AA(4401,100),CC(3,100)
*
  DATA MM/6*12,6*111,6*1101,6*4401/
  DATA NN/ 5, 6, 7, 8, 9,10, 8,18,28,38,48,98, -
           8,18,28,38,48,98, 8,18,28,38,48,98/
*
  T      = 3
  MO     = 12
  LDA    = 4401
  LDC    = 3
  LDQ    = 4401
  LDQ2   = 100
*
                                SET ORDINATES OF DATA
  YY(01)=2.2
  YY(02)=4.0
  YY(03)=5.0
  YY(04)=4.6
  YY(05)=2.8
  YY(06)=2.7
  YY(07)=3.8
  YY(08)=5.1
  YY(09)=6.1
  YY(10)=6.3
  YY(11)=5.0
  YY(12)=2.0
*
                                SET ABCISSAS OF DATA
  DO 10 I=1,MO
    XX(I)=2*I
10  CONTINUE
*
                                SET ABCISSAS OF CONDITIONAL DATA
  DX(1)= 6.0
  DX(2)=11.0
  DX(3)=19.0
*
  DO 100 KAI=1,24
*
  M      = MM(KAI)
  NBP    = NN(KAI)
  N      = NBP+2
  NMT    = N-T
*
  CALL GDATA(LDA,LDC,M,T,NBP,MO,XX,YY,DX,AA,CC,X1,W2,Y,CH)
*
  WRITE(6,(''1***** C1 *****''))
  DO 500 I=1,T
    WRITE(6,1000) CC(I,CH(1)),CC(I,CH(2)),CC(I,CH(3))
500  CONTINUE
1000  FORMAT(' ',3F10.3)

```

第 8 図 メインプログラムソースリスト (その 1)

```

*
WRITE(6,210) M,N
210 FORMAT(/14X,'シ"ヨウケン ツキ サイヨウ シ"シ"ヨウカイ (M =',I5,', N =',I3,',)')
220 FORMAT(1X,10F10.3)
*
*****
*
CALL ACYINT(M,N,T,LDA,LDC,AA,CC,Y,A,C,W1)
CALL CLOCKM(JIKAN1)
CALL PROJECT(LDA,LDC,LDQ,LDQ2,M,N,T,A,C,Q,Q2,X1,W1 )
CALL CLOCKM(JIKAN2)
TIME(KAI,1)=JIKAN2-JIKAN1
*
WRITE(6,'(/'O*** PROJECTION ***')')
WRITE(6,220) (X1(I),I=1,N)
*
CALL ACYINT(M,N,T,LDA,LDC,AA,CC,Y,A,C,W1)
CALL CLOCKM(JIKAN1)
CALL ELIMIN(LDA,LDC,LDQ,LDQ2,M,N,T,A,C,Q,X1,W1,W2,CH)
CALL CLOCKM(JIKAN2)
TIME(KAI,2)=JIKAN2-JIKAN1
*
WRITE(6,'(/'O*** ELIMINATION ***')')
WRITE(6,220) (X1(I),I=1,N)
*
CALL ACYINT(M,N,T,LDA,LDC,AA,CC,Y,A,C,W1)
CALL CLOCKM(JIKAN1)
CALL LAGRAN(LDA,LDC,LDQ,LDQ2,M,N,T,A,C,Q,Q2,X1,W1,W2)
CALL CLOCKM(JIKAN2)
TIME(KAI,3)=JIKAN2-JIKAN1
*
WRITE(6,'(/'O*** LAGRANGE ***')')
WRITE(6,220) (X1(I),I=1,N)
*** **
CALL ACYINT(M,N,T,LDA,LDC,AA,CC,Y,A,C,W1)
CALL CLOCKM(JIKAN1)
CALL PRONOR(LDA,LDC,LDQ,LDQ2,M,N,T,A,C,Q,Q2,X1,W1,W2)
CALL CLOCKM(JIKAN2)
TIME(KAI,4)=JIKAN2-JIKAN1
*
WRITE(6,'(/'O*** PROJECTION (NORMAL) ***')')
WRITE(6,220) (X1(I),I=1,N)
*
CALL ACYINT(M,N,T,LDA,LDC,AA,CC,Y,A,C,W1)
CALL CLOCKM(JIKAN1)
CALL ELINOR(LDA,LDC,LDQ,LDQ2,M,N,T,A,C,Q,X1,W1,W2,CH)
CALL CLOCKM(JIKAN2)
TIME(KAI,5)=JIKAN2-JIKAN1
*
WRITE(6,'(/'O*** ELIMINATION (NORMAL) ***')')
WRITE(6,220) (X1(I),I=1,N)
*
*****
*
100 CONTINUE
*

```

第 8 図 メインプログラムソースリスト (その 2)


```

WRITE(6, '(11/14X, 'シヨヨウ シカク ( タイ ミリSEC ) ' ) )
WRITE(6, '( / 'O( M , N )      PROJECTION      ELIMINATION      ' , -
      '      LAGRANGE      PROJECTION      ELIMINATION ' ) )
DO 20 KAI=1,24
  WRITE(6,200) MM(KAI),NN(KAI)+2,TIME(KAI,1),TIME(KAI,2), -
    TIME(KAI,3),TIME(KAI,4),TIME(KAI,5)
20 CONTINUE
200 FORMAT('O(',I4,',',I3,',',5(6X,I10))
*
STOP
END

```

第 8 図 メインプログラムソースリスト (その 3)

[6] サブルーチン副プログラムの説明

ここで、今回の数値実験のメインプログラム (第 8 図) で引用しているサブルーチン副プログラムについて説明する。

まず、5つの方法を連続してテストするために、問題の係数行列等を保存する必要がある、そのために大きな配列を二重にとらなければならなかったことをことわっておきたい。

そして、直接今回の目的ではないが、前処理として次の3つのサブルーチンを引用している。

" サブルーチン GDATA "

機能 : PROG 5 (参考文献 (4) の 27 章 4 節および付録

C) のデータから、補間により係数行列を作成する。

引用の形式 : CALL GDATA (LDA, LDC, M,
T, NBP, M0, XX, YY, XC, A, C, B,

X, Y, CH)

引数 : A 出力 係数行列 A

C 出力 条件行列 C

Y 出力 右辺のベクトル b

CH 出力 消去法で用いる列交換の指標

" サブルーチン ACYINT "

機能 : 係数行列, 条件行列, 右辺のベクトルを複写する.

引用の形式 : CALL ACYINT (M, N, T, LDA,
LDC, AA, CC, Y, A, C, W1)

引数 : A 出力 係数行列

C 出力 条件行列

W1 出力 右辺のベクトル

" 富士通サービスサブルーチン CLOCKM "

機能 : 実行可能プログラムの実行開始からのCPU占有時間

を, ミリ秒単位でIに返す.

引用の形式 : CALL CLOCKM (I)

さて, 今回比較の対象としたのは, 次の5つである.

" サブルーチン PROJECT "

機能 : Projection法により, 条件付最小自乗問題を解く.

ただし, 途中の条件なし最小自乗問題はQR法で解く.

引用の形式 : CALL PROJECT (LDA, LDC,
LDQ, LDQ2, M, N, T, A, C, Q, Q2,
X1, W1, W2)

" サブルーチン ELIMIN "

機能 : 消去法により, 条件付最小自乗問題を解く.

ただし, 途中の条件なし最小自乗問題はQR法で解く.

引用の形式 : CALL ELIMIN (LDA, LDC,
LDQ, M, N, T, A, C, Q, X1, W1, W2,
CH)

" サブルーチン LAGRAN "

機能 : Lagrange法により, 条件付最小自乗問題を解く.

引用の形式 : CALL LAGRAN (LDA, LDC,
LDQ, LDQ2, M, N, T, A, C, Q, Q2,
X1, W1, W2)

" サブルーチン PRONOR "

機能：Projection法により，条件付最小自乗問題を解く．

ただし，途中の条件なし最小自乗問題はNE法で解く．

引用の形式：CALL PRONOR (LDA, LDC,
LDQ, LDQ2, M, N, T, A, C, Q, Q2,
X1, W1, W2)

“ サブルーチン ELINOR ”

機能：消去法により，条件付最小自乗問題を解く．

ただし，途中の条件なし最小自乗問題はNE法で解く．

引用の形式：CALL ELINOR (LDA, LDC,
LDQ, M, N, T, A, C, Q, X1, W1, W2,
CH)

以上5つのサブルーチンの引数について，まとめて説明する．

LDA 入力 行列Aの第1寸法宣言子

LDC 入力 行列Cの第1寸法宣言子

LDQ 入力 行列Qの第1寸法宣言子

LDQ2 入力 行列Q2の第1寸法宣言子

M 入力 式の数m

N 入力 変数の数n

T 入力 条件の数 t
 A 入力 $m \times n$ の係数行列 (保存されない)
 C 入力 $t \times n$ の条件の係数行列 (保存されない)
 Q 作業 $m \times n$ の行列
 Q 2 作業 $n \times n$ の行列
 X 1 出力 n 次の解のベクトル
 W 1 入力 m 次の右辺のベクトル (保存されない)
 W 2 作業 m 次のベクトル
 C H 入力 t 次のベクトル (消去法のみ)

いずれのサブルーチンにおいても、三角連立1次方程式は、
 LINPACKのサブルーチンSTRSLを用いて解いてい
 る。また、NE法におけるCholesky分解は、同じくLIN-
 PACKのサブルーチンSCHDCを用いている。ただし、
 列ピボットを実行していないので、検討の余地がある。

さて、QR分解にはすべて行毎のGivens変換を用いたが、
 その典型的な例として、条件なし最小自乗問題を解く部分で
 引用しているサブルーチンGVNST2を示す。(第9図)

```

SUBROUTINE GVNST2(LDA,M,N,A,B)
INTEGER LDA,M,N
REAL A(LDA,1),B(1),COS,SIN
*
*   *** DECOMPOSE A = Q * R ( R-->A ) ***
*
*   *** SET C = TR(Q) * B ( C-->B ) ***
*
*   GIVEN'S TRANSFORM.
*
DO 30 I=2,M
  L=MIN(I-1,N)
  DO 20 J=1,L
    IF(A(I,J).EQ.0.0) GOTO 20
    CALL G1(A(J,J),A(I,J),COS,SIN,A(J,J))
    A(I,J)=0.0
    DO 10 K=J+1,N
      CALL G2(COS,SIN,A(J,K),A(I,K))
10    CONTINUE
      CALL G2(COS,SIN,B(J),B(I))
20    CONTINUE
30 CONTINUE
*
RETURN
END

```

第9図 サブルーチンGVNST2ソースリスト

" サブルーチン GVNST2 "

機能：行列をQR分解し，別のベクトルに左からQの転置行列をかけた積と，Rを出力する．

引用の形式：CALL GVNST2 (LDA, M, N, A, B)

引数：A 入力 $m \times n$ の行列A

出力 上三角部分にR

B 入力 m次のベクトルb

出力 $Q^T b$

ここで引用しているサブルーチン G_1 , G_2 は, ともに参考文献 (4) の 10 章および付録 C にあるもので, G_1 は 2 数に対して Givens 変換を構成して適用し, G_2 は構成された変換を他の 2 数に適用する.

ここでは, 変換の度に $A_{ij} = 0$ とおいているが, 本来その必要はない. また, Projection 法と Lagrange 法においては, それぞれ C^T または K の QR 分解の変換行列 Q (またはその一部) を残す必要があることに注意されたい.

[7] まとめ

- (1) 条件つき最小自乗法のアルゴリズムの中では, Lagrange 法が演算時間, 計算精度ともにすぐれている.
- (2) 正規方程式法は QR 法にすべての点で劣っている.
- (3) sparse な最小自乗法問題では, QR 法の演算時間は n のオーダーである.

謝辞 本研究のプログラムの作成にあたり, 独協大学情報センター 杉山武司氏に大変お世話になりました. ここに謝意を表します.

参考文献

- 1) G e o r g e, A. a n d H e a t h, M.
Linear Algebra and Its Applications, 34, (1980)
- 2) G e o r g e, A. a n d H e a t h, M.
Union Carbide Tech. Report ORNL/CSD-87 (1981)
- 3) C o l e m a n, T. F.
Large Sparse Numerical Optimization, Springer (1984)
- 4) L a w s o n, C. L. a n d H a n s o n, R. J.
Solving Least Squares Problems, Prentice-Hall (1974)