数理解析研究所講究録
第548巻 1985年 90-121

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ＳＡＬＳ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ＱＲ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ｆｉｔする問題である．この場合，係数行列Ａは第１図のようなブロ

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

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

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

$L y = A^T b$，$L^T x = y$ 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×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．

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

第３図　　　　　　　　　　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×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×ｎの下台形行列，

Ｕ：ｎ×ｎの上三角行列　である．

ｂ．Ｐｂ＝ｄ　とおいて

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～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．

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

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

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．

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

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）に表わした方が良い．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. $\overline{b} \leftarrow b - A y_1$

e. AZをつくる.

f. $\min \| (AZ) \overline{y}_2 - \overline{b} \|$ を解く.

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

g. $x \leftarrow y_1 + y_2$

(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．

$$\min \| \overline{A2}\, x_2 - \overline{b} \|$$

ここに $\overline{A2} \leftarrow A2 - A1\ C1^{-1} C2$，

$$\overline{b} \leftarrow b - A1\ C1^{-1} d$$

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．

$$Z = \begin{pmatrix} -C1^{-1}C2 \\ \\ I \end{pmatrix}, \quad y_1 = \begin{pmatrix} C1^{-1}d \\ \\ 0 \end{pmatrix}$$

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．条件式の数が少い

　　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．

$$\min \ 1/2\,\{x^T (A^T A)\, x - (A^T b)^T x\},\quad Cx = d$$

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．$(A^T A)\, y = A^T b$,

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

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

しかしHeath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．$A = Q\,(R,\ 0)^T$ と分解する．

ここに　$Q : m \times m$，$R : n \times n$

b．連立方程式　$Ry = c$ を解く（$c = Q^T b$）

c．tコの連立方程式

$$R^T K = C^T$$ を解く．

d．$K = \tilde{Q} L^T$　とQR分解する．

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

e．$r \leftarrow d - Cy$

f．連立方程式 $Ls = r$　を解く．

g．$v \leftarrow \tilde{Q} s$

h．連立方程式　$Rz = v$　を解く．

i．$x \leftarrow y + z$

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storge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．

$$
\begin{matrix}
\ast & \ast & \ast & \ast & 0 & 0 & 0 \\
\ast & \ast & \ast & \ast & \ast & 0 & 0 \\
\ast & \ast & \ast & \ast & \ast & \ast & 0 \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast \\
0 & \ast & \ast & \ast & \ast & \ast & \ast \\
0 & 0 & \ast & \ast & \ast & \ast & \ast \\
0 & 0 & 0 & \ast & \ast & \ast & \ast
\end{matrix}
$$

第５図　$A^T A$の形

$$
\begin{matrix}
\ast & \ast & \ast & \ast & 0 & 0 & 0 \\
0 & \ast & \ast & \ast & \ast & 0 & 0 \\
0 & 0 & \ast & \ast & \ast & \ast & 0 \\
0 & 0 & 0 & \ast & \ast & \ast & \ast \\
0 & 0 & 0 & 0 & \ast & \ast & \ast \\
0 & 0 & 0 & 0 & 0 & \ast & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & \ast
\end{matrix}
$$

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．Ｐａｒｔｅｒによれば，

（Ｌｅｍｍａ）$B_{IJ} = 0$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．

*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*
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７図　直交行列Ｑの形

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くするような処理をしていない．

［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ｍｓｅｃ）

(ｍ，ｎ)	ＰＲＯ	ＥＬＩ	ＬＡＧ	＊ＰＲＯ	＊ＥＬＩ
(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) ＰＲＯ，ＥＬＩ，ＬＡＧ：それぞれProjection法，消去法，Lagrangeの乗数法

3) ＊はＱＲ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ＱＲ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になっているか推測することが出来る．

(m, n)	PRO	ELI	LAG	*PRO	*ELI
(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 *****
    -0.479      0.005      0.0
     0.0       -0.001      0.0
     0.0        0.0       -0.432

             ｼﾞｮｳｹﾝ ﾂｷ ｻｲｼｮｳ ｼﾞｼﾞｮｳｶｲ (M =  111 , N = 20)

*** PROJECTION ***
     0.935     1.398     2.302     2.910     3.442     3.065     3.035     1.796     1.831     1.792
     2.449     2.917     3.495     3.948     4.257     4.076     3.731     2.754     1.174     0.628

*** ELIMINATION ***
     0.942     1.397     2.303     2.911     3.442     3.065     3.031     1.797     1.831     1.793
     2.449     2.918     3.495     3.948     4.257     4.076     3.731     2.754     1.174     0.628

*** LAGRANGE ***
     0.936     1.398     2.302     2.910     3.442     3.065     3.035     1.796     1.831     1.792
     2.449     2.918     3.495     3.948     4.257     4.076     3.731     2.754     1.175     0.628

*** PROJECTION (NORMAL) ***
     0.935     1.398     2.302     2.910     3.442     3.065     3.035     1.796     1.831     1.792
     2.449     2.917     3.495     3.948     4.257     4.076     3.731     2.754     1.174     0.628

*** ELIMINATION (NORMAL) ***
     6.334     0.170     2.856     4.057     3.334     3.903    -4.040    17.223   -17.110    23.786
     2.016    -0.440     1.141    -0.606     1.113     0.068     0.541     0.416    -0.463     6.041
```

第４表　精度（m＝１１１，n＝２０）

20

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

            ジョウケン ツキ サイショウ ジジョウカイ (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）

```
***** C1 *****
    -0.126     -0.208      0.0
     0.0       -0.004      0.179
     0.0        0.0       -0.082

             ｼﾞｮｳｹﾝ ﾂｷ ｻｲｼｮｳ ｼﾞｼﾞｮｳｶｲ (M = 4401 , N = 10)

*** PROJECTION ***
    10.229     -0.552      4.186      2.446      1.558      3.058      4.165      4.167      1.924     -4.535

*** ELIMINATION ***
    10.459     -0.540      4.178      2.446      1.560      3.056      4.165      4.166      1.931     -4.571

*** LAGRANGE ***
    10.208     -0.552      4.186      2.446      1.557      3.058      4.164      4.166      1.928     -4.554

*** PROJECTION (NORMAL) ***
    10.170     -0.545      4.182      2.447      1.552      3.063      4.160      4.163      1.930     -4.561

*** ELIMINATION (NORMAL) ***
  -244.113     43.948     -6.909     14.071    -19.348     35.034      7.182      8.473     53.185   -347.480
```

第６表　精度　（ｍ＝４４０１，ｎ＝１０）

```
************************************************************************
****  ｼﾞﾖｳｹﾝ ﾂｷ LS ﾓﾝﾀﾞｲ ( ｼﾖﾖｳ ｼﾞｶﾝ ﾉ ﾋｶｸ )  T E S T ﾒｲﾝ ﾌﾟﾛｸﾞﾗﾑ   ****
************************************************************************
*
      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,'ｼﾞｮｳｹﾝ ﾂｷ ｻｲｼｮｳ ｼﾞｼﾞｮｳｶｲ (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 PROJCT(LDA,LDC,LDQ,LDQ2,M,N,T,A,C,Q,Q2,X1,W1    )
      CALL CLOCKM(JIKAN2)
      TIME(KAI,1)=JIKAN2-JIKAN1
*
      WRITE(6,'(/''0*** 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,M,N,T,A,C,Q,X1,W1,W2,CH)
      CALL CLOCKM(JIKAN2)
      TIME(KAI,2)=JIKAN2-JIKAN1
*
      WRITE(6,'(/''0*** 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,'(/''0*** 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,'(/''0*** 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,M,N,T,A,C,Q,X1,W1,W2,CH)
      CALL CLOCKM(JIKAN2)
      TIME(KAI,5)=JIKAN2-JIKAN1
*
      WRITE(6,'(/''0*** 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,'(''1''/14X,''ｼﾖﾖｳ ｼﾞｶﾝ ( ﾀﾝｲ ﾐﾘSEC )'')')
      WRITE(6,'(/''0(  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('0(',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り，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：PROG5（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　ＰＲＯＪＣＴ〃

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　ＰＲＯＪＣＴ（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×ｎの係数行列（保存されない）

Ｃ　入力　ｔ×ｎの条件の係数行列（保存されない）

Ｑ　作業　ｍ×ｎの行列

Ｑ２　作業　ｎ×ｎの行列

Ｘ１　出力　ｎ次の解のベクトル

Ｗ１　入力　ｍ次の右辺のベクトル（保存されない）

Ｗ２　作業　ｍ次のベクトル

ＣＨ　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×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Ｇ１，Ｇ２は，ともに参考文献（４）の１０章および付録Ｃにあるもので，Ｇ１は２数に対してGivens変換を構成して適用し，Ｇ２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Ａij＝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) George, A. and Heath, M.
Linear Algebla and Its Applications, 34,(1980)

2) George, A. and Heath, M.
Union Carbide Tech. Report ORNL/CSD-87 (1981)

3) Coleman, T. F.
Large Sparse Numerical Optimization, Springer (1984)

4) Lawson, C. L. and Hanson, R. J.
Solving Least Squares Problems, Prentice-Hall (1974)