

# 高次レオンチエフ行列の逆転について

多 賀 保 志  
藤 原 長 司

(1958年2月受付)

## On Inversion of the Leontief Matrices of Higher Order

Yasushi TAGA and Chodi HUDIWARA

In the present paper the various methods of inversion of the Leontief matrices of higher order will be investigated.

By the universal automatic relay computer in our institute, we computed the inverse matrices of two Leontief matrices of which orders are 60 and 32 respectively.

The method of Verzuh and that of partition were used for the inversion of the former matrix, and the method of Verzuh was adopted for the latter. It took 36 hours by the method of partition and 4.25 hours by the method of Verzuh. If the former matrix is treated simply by the method of Verzuh, it will take 32 hours by our practical considerations.

The computing errors of the inverse of a matrix might be considered to be increased with the order of matrices. But, our actual results assured considerably high accuracy in inversion of Leontief matrices. The computing errors of inverse matrices can be roughly evaluated by the inequality,

$$\rho(E) \leq \rho(\tilde{A}^{-1}) \frac{\rho(R)}{1 - \rho(R)}$$

in which  $\rho$  represents the norm of matrices.

In our results,  $\rho(E) < 3.914 \times 10^{-5}$  when  $n=60$ . But it has been treated in several papers, however it may be in same restricted situations, that the errors are not increased in the Leontief matrix with the order.

Thus, we are encouraged to be able to compute the inverse of Leontief matrices of higher order than 60 with high accuracy.

The Institute of Statistical Mathematics

### §0. 序 論

近時、投入産出分析 (in-input out-put analysis) と呼ばれる方法が国民経済の分野における複雑な産業構造を解明する手段として用いられる様になり、最近はとくに多部門にわたる産業連関

表 (Leontief table) を取扱う必要が生じてきている。その場合、種々な数値計算が要求されるが、そのうちでとくにレオンチエフ行列の逆転が最も必要かつ困難なものとしてされている。

従来の電動計算機で行列の逆転を行う場合、高々 20 次までであり、それ以上の高次逆行列を求める事は計算量が莫大のためほとんど不可能視されていたが、大型自動計算機の出現によつてそのような計算も容易に遂行出来る様になった。

我々は当研究所のリレー式万能型自動計算機 (FACOM-128) により、32 次および 60 次のレオンチエフ行列の逆転を試みたが、その方法の概要を述べて見よう。

尚本稿を書くに当つて、通商産業大臣調査統計部解析課、寺西武美氏、経済企画庁調査部統計課、楠田 義氏の両氏ならびに、統計数理研究所の林 知己夫氏、鈴木雪夫氏の両氏に種々助言を戴いた。又丸山愛子嬢には計算遂行にあたり、一方ならぬお世話になつた。上記の方々には厚く感謝する次第です。

### §1. 計算方法の選定

計算の実施にあたり、先ず産業連関表より、各要素を総産出額で割つた投入係数の行列  $A$  を作り、ついでレオンチエフ行列 ( $I-A$ ) を求めるため、若干の準備計算が必要である。(ここでは単位行列  $I$  より係数行列  $A$  を引いた ( $I-A$ ) をレオンチエフ行列と呼ぶことにする)。

その準備に要した時間をまとめてみると、次のようになる(テープ作成のパンチに要する時間は個人差が著しいが、我々がパンチに要した時間である)。

次数	テープ作成 (パンチ)	check	投入係数の 計	check	計
32	200 分	5 分	13 分	5 分	約 4 時間
60	720 分	15 分	45 分	12 分	約 13 時間

さて実際の分析において輸入係数とくに競走輸入係数の安定性が問題となる(文献 [2] 参照の事)。産業連関表において、輸入を表現する形式は種々考えられているが、一度逆行列を計算してから輸入係数のみを変えたい場合は、近似逆行列が求められたものとして、逐次近似法により求める事が出来る。このように、行列の投入係数を変更した上で、改めて逆行列を求める方法としては逐次近似法によつて行うことが出来るが、計算結果への影響度は比較的小さいのが普通である(これはいわゆるデータ誤差による影響とみなしてもよく、その場合の誤差評価の方法は文献 [3] に詳しく述べられている)。

以後、( $I-A$ ) を改めて  $A$  とよぶことにする。高次レオンチエフ行列の逆転には種々の方法があるが、計算の精度・所要時間等の点で、従来の経験より Verzuh の消去法が最も能率的であることがわつていたので、それを採用することにした。ただし 60 次の行列を逆転するには、約 14 巻 (1,400 m) のテープが必要となるので、それを節約するために分割法を併用することにした。60 次の行列を分割するには、いろいろの仕方があるが、計算量を少なくし、プログラムを単純化するには、4 つの 30 次行列に分割するのが最良であると考えた(文献 [1] 参照のこと)。これについて計算の手順を示すと次のようになる。

$$A = \left( \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) \text{より} \quad A^{-1} = \left( \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) \text{の形で求められたとすると,}$$

- 1°  $A_{11}^{-1}$  (Verzuh の方法による逆転)
- 2°  $X = A_{11}^{-1} \cdot A_{12}$  (行列の積)
- 3°  $Y = A_{21} \cdot A_{11}^{-1}$  (同 上)
- 3°  $Y \cdot A_{12}$  (同 上)

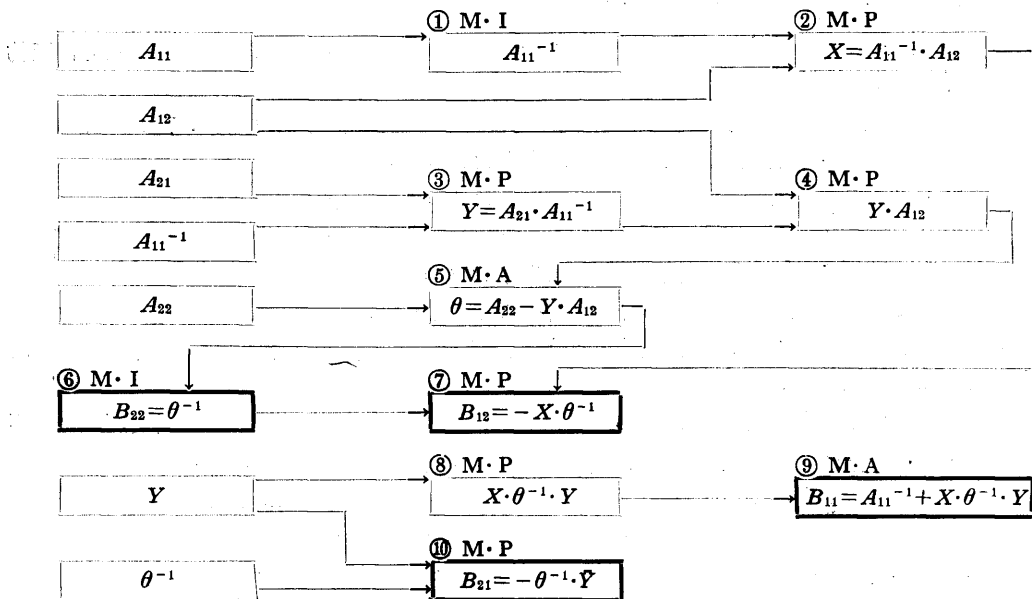
- 5°  $\theta = A_{22} - Y \cdot A_{12}$  (行列の減算)
- 6°  $B_{22} = \theta^{-1}$  (Verzuh の方法による逆転)
- 7°  $B_{12} = -X \cdot \theta^{-1}$  (行列の積)
- 8°  $X \cdot \theta^{-1} \cdot Y$  (同上)
- 9°  $B_{11} = A_{11}^{-1} + X \cdot \theta^{-1} \cdot Y$  (行列の和)
- 10°  $B_{21} = -\theta^{-1} \cdot Y$  (行列の積)

となる。

in-input, および out-pup にはテープを用いるが, 行列はテープの上では次元の表現しかできないから, 行列の積演算を行うに当り, 左因子は行ベクトルを, 右因子は列ベクトルを順次にパンチしておくことが大切である。また, その結果が行ベクトルとして表わされた方がよいか, 列ベクトルとして表わされた方がよいか, という点に十分注意して, 演算操作を行うように注意すべきであろう。

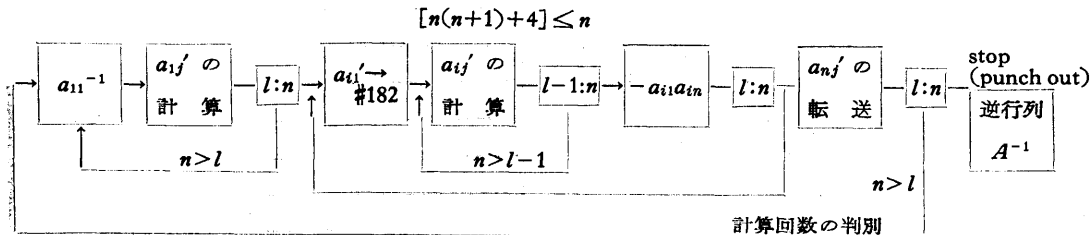
以上の演算過程を flow-chart で示せば Fig. 1.1 のようになる (なお, 次数が 10 以下の flow-chart は Fig 1.2 に示した)。

Fig 1.1 分割法に依る逆行列の flow-chart



M·I : 行列の逆転用ルーティン      ①~⑩: 番号は前頁計算の手順参照。  
 M·P : 行列の積用ルーティン  
 M·A : 行列の加(減)法用ルーティン

Fig 1.2 内部記憶のみによる逆行列の flow-chart



$n$ : matrix of order  
 $n(n+1)+4$ : inner storage のみ用いる時, 必要な storage の数,  $\det |A|$  も同時に計算される。

以上分割法によつて行列を逆転するに要する計算をまとめてみると、

{	小行列の逆転	2 回
	小行列の積	6 回
	小行列の加算 (又は減算)	2 回

となる。

但し、産業連関表より、レオンチエフ行列を作り、一貫して計算機を用いるならば、転置行列を作る操作がこの外に 3 回必要である。これを避けるには、元の小行列  $A_{21}$  を初めから転置行列  $A_{21}'$  としておけば操作が 1 回少なくなる。

§2. 計算の実施

もう一度はじめから、分割法による計算手順および所要時間をまとめてのべてみると、次のようになる: ( $n=60$  として)

[i]	産業連関表のテープ作成	720 分
[ii]	投入係数の計算	45 分
[iii]	レオンチエフ行列の作成	20 分
[iv]	レオンチエフ行列の逆転	2200 分

となり、合計所要時間は約 50 時間であつた (ただし、実際には途中で機械の故障等のため、延約 60 時間を要した)。

この中では、[iv] の段階が最も複雑で長時間を必要とするが、その内訳を述べてみると、

{	小行列の逆転	255 分 × 2 = 510 分
	小行列の積	260 分 × 6 = 1,560 分
	小行列の加(減)算	5 分 × 2 = 10 分
	小行列の転置	30 分 × 3 = 90 分

となつており、小行列の乗算に大部分の時間を使うことがわかる。

もし、これを分割法によらず、単純な Verzhuh 法によつて一貫操作でやると所要時間は約 32 時間に短縮されるはずである。

また、32 次の行列の逆転を、やはり単純な Verzhuh 法によつて行つたところ、所要時間は約 5

Fig. 2・1 行列の次数  $n$  と逆行行列算出所要時間の関係 ( $n > 10$ )

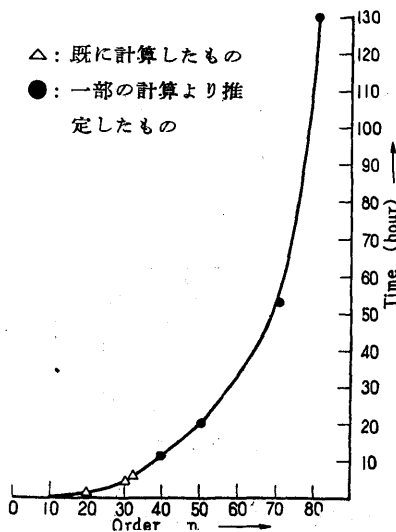
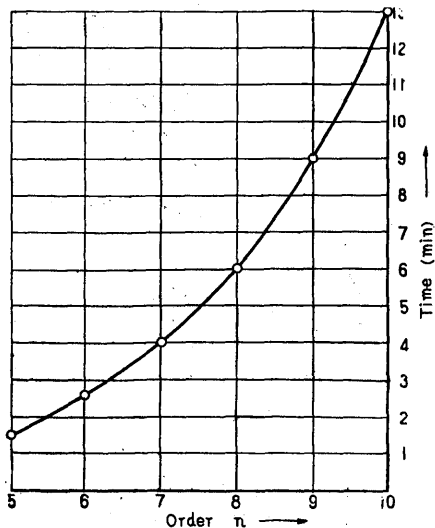


Fig. 2・2 行列の次数と逆行行列算出所要時間の関係 ( $n \leq 10$ )



時間 15 分であつた。

以上の結果から、行列の次数  $n$  とその逆転に要する時間の関係を図示してみると、Fig 2.1 のようになり、これは三次曲線で近似される。 $n \leq 10$  なる場合も別の機会に試みたのでその結果を表示すると、Fig. 2.2 のようになる。

### §3. 計算の精度

計算開始にあつては、高次行列の逆転に際して、計算結果の精度が一般に非常に悪くなるのではないかと心配された。しかし、実際には下に述べる如く、意外に精度のよい結果がえられたが、これはレオンチエフ行列の特殊な性質によるものと考えられる。

精度をみる一方法として行列  $(I - A^{-1} \cdot A)$  を作つて、その要素の絶対値の大きさを眺めて見ると、Table 3. の様になる。但し行列の積  $A^{-1} \cdot A$  を求めるには、逆転とほぼ同等の時間を要するので、部分的に検討してみた。即ち元の行列  $A$  よりいくつかの列ベクトル  $\mathcal{X}_j$  を適当に選び、逆行列  $A^{-1}$  との積  $A^{-1} \cdot \mathcal{X}_j$  を作つて、その第  $j$  要素が 1、その他の要素が 0 となるか否かをみた。

このような残差行列による評価は普通に行われているが、ここではさらに計算されてえられた逆行列そのものの誤差評価を試みよう。

$A$ : 元の行列

$A^{-1}$ :  $A$  の真の逆行列

$\tilde{A}^{-1}$ : 算出された  $A$  の近似逆行列

$E$ : 誤差行列  $\tilde{A}^{-1} - A^{-1}$

$R$ : 残差行列  $A \cdot \tilde{A}^{-1} - I$

とすれば、
$$\tilde{A}^{-1} = A^{-1} + E, \quad R = A \cdot \tilde{A}^{-1} - I$$
なる関係から

$$\begin{aligned} R &= A \cdot \tilde{A}^{-1} - I = A \cdot E \\ E &= A^{-1} \cdot R = \tilde{A}^{-1} R (I + R)^{-1} \end{aligned}$$

なる関係がえられる。

ここで  $R$  のノルムが 1 より小ならば、

$$E = \tilde{A}^{-1} \{ R - R^2 + R^3 \cdots + (-1)^{n+1} R^n \cdots \}$$

となる。これで誤差行列  $E$  のノルムは勿論、各要素の絶対値（すなわち近似逆行列  $\tilde{A}^{-1}$  の各要素の絶対誤差）を精密に知る事が出来る。

レオンチエフ行列の逆転に於ては、 $R$  は一般に十分小さいから (Table 3 参照)、次の近似で十分である。

$E$  の第 1 近似:  $\tilde{A}^{-1} \cdot R$

又は、 $E$  の第 2 近似:  $\tilde{A}^{-1} \cdot R - \tilde{A}^{-1} R^2$

次にノルムで評価すれば、

$$\rho(E) \leq \rho(\tilde{A}^{-1}) \{ \rho(R) + \rho(R^2) + \cdots + \rho(R^n) + \cdots \} = \rho(\tilde{A}^{-1}) \cdot \frac{\rho(R)}{1 - \rho(R)}$$

となることは容易にわかる。

$E = (e_{ij})$  とすれば

$$\max |e_{ij}| \leq \rho(E) \leq \rho(\tilde{A}^{-1}) \cdot \frac{\rho(R)}{1 - \rho(R)}$$

となつて、このような評価は多少粗くなることは止むをえない。60 次の行列の逆転の際の誤差をノルムによつて評価してみよう。行列の積は前記理由により長時間の演算を要するので、60 の行

Table 3. 算出逆行列 check ( $A^{-1} \cdot A = I$ ) の一部 (column vector)

n = 10		n = 32		n = 60	
$A^{-1} \cdot \mathcal{A}_1'$	$A^{-1} \cdot \mathcal{A}_{10}'$	$A^{-1} \cdot \mathcal{A}_1'$	$A^{-1} \cdot \mathcal{A}_{32}'$	$A^{-1} \cdot \mathcal{A}_1'$	$A^{-1} \cdot \mathcal{A}_{60}'$
0.99999996	0	1.0000001	0.0 <sup>8</sup> 1	1.0000001	-0.0 <sup>8</sup> 2
-0.0 <sup>8</sup> 27	-0.0 <sup>7</sup> 2	-0.0 <sup>8</sup> 6	-0.0 <sup>8</sup> 1	0.0 <sup>8</sup> 571	0
-0.0 <sup>7</sup> 17	0.0 <sup>7</sup> 1	-0.0 <sup>8</sup> 38	-0.0 <sup>8</sup> 1	-0.0 <sup>8</sup> 898	-0.0 <sup>8</sup> 3
-0.0 <sup>7</sup> 36	0	-0.0 <sup>8</sup> 14	0.0 <sup>8</sup> 2	-0.0 <sup>7</sup> 31048	0
-0.0 <sup>8</sup> 19	0	-0.0 <sup>8</sup> 9	0.0 <sup>7</sup> 4	0.0 <sup>7</sup> 11406	-0.0 <sup>8</sup> 2
0.0 <sup>8</sup> 9	0	-0.0 <sup>8</sup> 16	-0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 3352	-0.0 <sup>8</sup> 6
-0.0 <sup>8</sup> 26	0.0 <sup>7</sup> 1	-0.0 <sup>8</sup> 9	0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 117	0.0 <sup>8</sup> 1
0.0 <sup>8</sup> 9	0	0	0.0 <sup>7</sup> 3	0.0 <sup>8</sup> 66	-0.0 <sup>8</sup> 5
-0.0 <sup>8</sup> 16	0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 9	0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 19	-0.0 <sup>8</sup> 1
-0.0 <sup>8</sup> 2	0.99999999	-0.0 <sup>8</sup> 1	0	-0.0 <sup>10</sup> 5	-0.0 <sup>8</sup> 1
		-0.0 <sup>8</sup> 74	0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 142	-0.0 <sup>8</sup> 4
		-0.0 <sup>8</sup> 9	0.0 <sup>8</sup> 3	-0.0 <sup>8</sup> 594	-0.0 <sup>8</sup> 1
		0.0 <sup>8</sup> 14	0	0.0 <sup>8</sup> 1257	-0.0 <sup>8</sup> 2
		0.0 <sup>8</sup> 5	0.0 <sup>7</sup> 2	-0.0 <sup>8</sup> 1586	0
		0.0 <sup>8</sup> 3	0	-0.0 <sup>8</sup> 3	-0.0 <sup>8</sup> 3
		-0.0 <sup>8</sup> 79	-0.0 <sup>7</sup> 3	⋮	⋮
		0.0 <sup>8</sup> 77	0.0 <sup>8</sup> 8	⋮	⋮
		0.0 <sup>8</sup> 29	0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 194	-0.0 <sup>8</sup> 4
		0.0 <sup>8</sup> 13	0.0 <sup>7</sup> 3	0.0 <sup>8</sup> 675	-0.0 <sup>8</sup> 1
		-0.0 <sup>8</sup> 3	0.0 <sup>7</sup> 1	0.0 <sup>8</sup> 154	-0.0 <sup>8</sup> 4
		0	0.0 <sup>7</sup> 8	-0.0 <sup>8</sup> 83	-0.0 <sup>8</sup> 4
		-0.0 <sup>8</sup> 12	0.0 <sup>7</sup> 4	0.0 <sup>8</sup> 2036	0
		0.0 <sup>8</sup> 1	-0.0 <sup>7</sup> 1	-0.0 <sup>8</sup> 2122	-0.0 <sup>8</sup> 1
		-0.0 <sup>8</sup> 39	0	0.0 <sup>7</sup> 15611	-0.0 <sup>8</sup> 2
		-0.0 <sup>8</sup> 5	-0.0 <sup>7</sup> 1	-0.0 <sup>8</sup> 277	-0.0 <sup>8</sup> 3
		-0.0 <sup>8</sup> 2	-0.0 <sup>7</sup> 2	0.0 <sup>8</sup> 11	-0.0 <sup>7</sup> 2
		-0.0 <sup>8</sup> 5	0.0 <sup>8</sup> 2	-0.0 <sup>8</sup> 683	-0.0 <sup>8</sup> 4
		-0.0 <sup>8</sup> 5	0.0 <sup>8</sup> 3	-0.0 <sup>8</sup> 155	-0.0 <sup>8</sup> 7
		-0.0 <sup>8</sup> 21	0	-0.0 <sup>8</sup> 685	-0.0 <sup>8</sup> 1
		-0.0 <sup>8</sup> 2	0.0 <sup>8</sup> 6	0.0 <sup>8</sup> 2625	-0.0 <sup>8</sup> 1
		-0.0 <sup>8</sup> 63	0.0 <sup>7</sup> 4	-0.0 <sup>8</sup> 391	-0.0 <sup>8</sup> 2
		-0.0 <sup>8</sup> 3	1.0000000	0.0 <sup>8</sup> 6	0.99999990

註 1. 60 元は途中の element を省略した。  
 2. -0.0<sup>8</sup>27 は -0.0000000027 を示す, 以下同様。

と列より random に 10 行, 10 列を選んだ。

$\hat{\rho}^2(R)$  を  $\rho^2(R)$  の推定値とすれば, サンプリングの理論よりよく知られているように,

$$\hat{\rho}^2(R) = \frac{n^2}{m^2} \sum_{k,l} |r_{ikjl}|^2$$

としてよい。

ここで  $n$  は元の行列の次数,  $m(m < n)$  は  $n$  より random に選ばれた行及び列ベクトルの数であつて, 我々は  $n=60, m=10$  とした。

これより  $\rho^2(R)$  の上限を押えるには,

$$\hat{\rho}^2(R) - 3\sigma_{\hat{\rho}^2(R)} \leq \rho^2(R) \leq \hat{\rho}^2(R) + 3\sigma_{\hat{\rho}^2(R)}$$

が殆んど確実に成立つから ( $\sigma$  は  $R$  の要素の二乗の標準偏差),

$$\rho^2(R) \leq \hat{\rho}^2(R) + 3\sigma_{\hat{\rho}^2(R)}$$

として上限が知れる. したがつて,

$$\rho(E) \leq \rho(\tilde{A}^{-1}) \cdot \frac{\rho(R)}{1 - \rho(R)} \leq \frac{\sqrt{\hat{\rho}^2(R) + 3\sigma_{\hat{\rho}^2(R)}}}{1 - \sqrt{\hat{\rho}^2(R) + 3\sigma_{\hat{\rho}^2(R)}}} \rho(\tilde{A}^{-1})$$

によつて誤差行列のノルムが評価できる.  $n = 60$  の時,  $\rho(E) \leq 3.914 \times 10^{-5}$  であつた.

計算の精度は, 一般に次数が多くなるに従つて悪くなるものであるが, レオンチエフ行列の場合次数にはあまり影響がないようにみえることは, さらに高次 (たとえば 100 次以上) のレオンチエフ行列の逆転を行う場合にも明るい見通しを与えているといえよう.

#### §4. 結 語

以上高次レオンチエフ行列の逆転について述べてきたが, 最後に計算方法の選定, 計算実施上の注意, programming を行う時の問題点等についてふれてみたい.

まず計算方法の選定には, 上に述べたように, 計算の速度・精度の点からみて, Verzuh 法によるのが最良であろう. そこで問題となるのは, 分割法を採用するか否かである. 計算時間の点からいえば, 分割法を用いた場合も, 用いない場合も, ほぼ同じと考えて差支えない (前者の方が約 2 割位余計に時間がかかる). しかし, 途中で何度も hand operation (数値テープと命令テープのかけかえ等) が入るため, operation error を誘発しやすいのが欠点である. そのような error は, 途中で一度入り込むと, 最後まで影響するし, またそれを発見することは, 最終段階に到るまで非常にむづかしい. したがつて, できる限り分割法によらぬ一貫操作を採用することが最も望ましい. もしテープの消費量を減らすためとか, 記憶容量の関係で, どうしても分割法を採用せざるを得ない場合には, 各段階毎に計算の check を厳重に行い (今のところ, 2度ずつ同じ operation を行う以外に適当な check 法がない), 慎重に計算を進めて行くことが大切である. 又何時停電等の予期しない事故が起るかも知れないから, 如何なる step で停止しても, その step からやり直しができるような program を用いるべきである. 分割法によるか否かは, 計算実施に当り以上の点を考慮して決定すべきであるが, FACOM-128 によつて計算する時は我々の経験によれば,  $n \leq 50$  ならば一貫操作を,  $n > 50$  ならば分割法を採用するのがよいと考えられる. とくに,  $n \leq 10$  なる場合は, テープ記憶によらず, in-put と out-put 以外は, すべて内部記憶によつて処理することも出来るが, その Program を附表 3 に掲げた (flow-chart Fig 1.2 参照).

これによると, 計算前に, 一度行列を送り込むだけで, 以後人手を全く必要とせず, 途中のテープも要らないから次数が余り大きくない時有効であり, 将来高速度の計算機で, 大容量の memory を備えるものが出現したら偉力を発揮するであろう.

次表に各法に依る必要な memory の数とテープの使用量を記しておいた. ( $n$  は次数)

計 算 方 式	必要な memory の数	所要 Tape (m)
a) 分 割 法	$\frac{n}{4} + 2$	$3n^2(n + 27)/2,000$
b) Verzuh 法 (punch-out)	$n + 2$	$6n^2(n + 1)/1,000$
c) Verzuh 法 (memory のみによる)	$n(n + 1) + 4$	* * *

\* b), c) は  $\det |A|$  が同時に計算される.

\*\* 所要 Tape は in-put 及び out-put のみで足りる.

つぎに, **programming** を行う上の注意を述べてみよう. まず, 計算順序を示す **flow-chart** を描いて, それを忠実に **programming** すればよいわけであるが, 行列の積や, ベクトルの加減算を行うに当つて, 組込みルーティンを使用すべきか否か, という点に問題がある.

結論を述べるならば,  $n > 10$  の場合は, その次数  $n$  に応じた専用ルーティンを **program** に組んだ方が計算時間はずつと短縮されるから得であり,  $n \leq 10$  ならば組込みルーティンを使用した方が, **programming** およびテープ作成の手間が省けるから, 有利である. これは行列の逆転だけでなく, 他の計算においてもしばしば問題になるであろうが, 一般的にいうならば

- 1) **programming** に要する時間
- 2) **tape** 作成に要する時間
- 3) 計算所要時間

の三者の和が最小になるような方式を採用することが望ましいのである. ちなみに, もし 60 次の行列を, 組込みルーティンを使用して上記のような一貫操作を行うとすれば, 計算所要時間だけで実に 150 時間以上を要すると推定されることをみても, **programming** の重要性が理解される.

近い将来我が国においても, 100 次以上の行列を扱う必要が起ると思われるが, その場合にはとてもリレー計算機では処理できない程, 龐大な計算量の処理が要求されることになるであろう. アメリカにおいては, すでにそのような高次行列の逆転を, 電子計算機によつてやすやすと解いているが, 我が国においてもなるべく速かに, 優秀な電子計算機の出現が期待される所以である.

統計数理研究所

### 参 考 文 献

- 1) 多賀保志, “リレー計算機による線型計算について”, 統計数理研究所彙報, 5巻1号.
- 2) 通商産業大臣官房調査統計部編, “日本経済の産業連関分析”.
- 3) 林 知己夫, “レオンチエフ行列におけるデータ誤差の影響について”, 経済研究, 第7巻第2号, 一橋大学経済研究所編.
- 4) 森口繁一, “Leontief 行列の逆転について”, P. C. 資料 No. 17, 1955.
- 5) P. S. DWEYER, “Linear Computations”, Wiley 1951.
- 6) F. M. VERZUH, “The solution of simultaneous linear equations with the aid of the 602 Calculating Punch,” Table and Other Aids to Computation, Vol. 3, No. 27, July 1949.
- 7) F. V. WAUGH, “Inversion of the Leontief matrix by power series”, *Econometrica* 18, 142-154 (1950).
- 8) J. L. HOLLEY, “Note on the inversion of the Leontief matrix”, *Econometrica* 19, 317-320 (1951).
- 9) G. W. PETRIE, “Matrix inversion of simultaneous linear algebraic equations with the I. B. M. 604 Electric calculating punch”, 15\*.
- 10) J. TODD, “Experiments on the Inversion of a  $16 \times 16$  matrix,” 16\*.
- 11) J. SHERMAN, “Computations relating to inverse matrices”, 18\*.

\* “Simultaneous linear equations and the Determination of eigenvalues”, N. B. S. Applied mathematics series 29 (1953).



[附表 1] 行列の積用ルーティン (30 元専用)

計算式 $A(30.30) \cdot B(30.30) = C \cdot (30.30)$										INPUT	
R-1 (endless)										191 $a_1, a_2, \dots, a_{30}$ endless	
										192 $b_1', b_2', \dots, b_{30}'$	
初期条件 30 → #189 (SNI)											
ステップ	A	B	C	O			D			備考	
				1	2	3	1	2	3		
1	192	*	001	00	0	0	0	0	0	} $b_j'$ の read in	
2	192	*	002	00	0	0	0	0	0		
⋮	⋮	⋮	**	⋮	⋮	⋮	⋮	⋮	⋮		
⋮	⋮	⋮	**	⋮	⋮	⋮	⋮	⋮	⋮		
30	192	*	030	00	0	0	0	0	0		
31	030	*	*	13	1	1	1	0	0		30 → SNO (回転数記憶)
32	002	*	*	14	0	0	0	0	0		call R-2.
33	001	000	*	16	0	0	0	0	0		SNI の内容により call R-1 計算終了の時 call R-0 (操作台)

\* 印は計算に関係のない address で普通 000 とする。

\*\* 3 step より 29 step 迄は C-address のみ、ステップ数と同一の address にする。

R-2 (endless)

1	220	220	*	01	1	0	0	0	0	0 → #210
2	191	001	*	03	1	0	0	0	0	$a_i, b_j \rightarrow \#211$
3	210	211	*	01	1	0	0	0	0	} $\sum a_i b_j$
**	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
60	191	030	*	03	1	0	0	0	0	
61	210	211	191	01	0	0	0	0	0	$a_i, b_j \rightarrow \#191$ に punch out
62	002	001	*	15	0	0	0	0	0	call R-2 or R-1 by SNO

\*\* 4 step より 59 step 迄は、odd step はそのままの繰返しで、他は B-address のみを1づつあげる。

註 1. 60 元の逆転は、この routine と、[附表 2] を用いた。

尙、行列の加(減)算用 routine は省略した。

2. 行列の積 C は #191 より punch out される。

[附表 2] 逆行列用ルーティン (30 元専用)

計算式		初期条件							INPUT	備 考
R-1 (endless)		$0 \rightarrow \#182$ $1 \rightarrow \#184$							191 <span style="border: 1px solid black; padding: 0 2px;">A</span>	
ステップ	A	B	C	O			D			
				1	2	3	1	2	3	
1	221	182	182	01	0	0	0	0	0	計算回数 $N$
2	191	*	183	00	0	0	0	0	0	$a_{11} \rightarrow \#183$
3	183	184	184	03	0	0	0	0	0	$\det  A $ の計算
4	*	*	*	26	1	0	1	1	1	$N, a_{11}, \det  A $ の Print
5	221	183	030	04	0	0	0	0	0	$a_{11}^{-1} \rightarrow \#30$
6	191	030	001	03	0	0	0	0	0	$a_{11}^{-1} \cdot a_{1j}$
	⋮	⋮	**	⋮	⋮	⋮	⋮	⋮	⋮	$a_{ij}$ の計算
34	191	030	029	03	0	0	0	0	0	
35	029	*	*	13	1	1	1	0	0	29 $\rightarrow$ SNO
35	002	*	*	14	0	0	0	0	0	call R-2
37	000	*	*	13	1	0	1	0	0	$0 \rightarrow \#195$ (SA-A)
38	030	*	*	13	1	1	1	0	0	30 $\rightarrow$ SNO
39	090	*	*	14	0	0	0	0	0	call R-90
40	*	*	003	25	0	1	0	0	0	穿孔機 3 step jump
41	*	*	003	24	0	1	0	0	0	読取機 3 step jump
42	001	*	*	14	0	0	0	0	0	call R-1

\*\* 7 step より 33 step 迄は, C-address のみを, 1 づつあげる.

R-2 (endless)										
1	191	*	000	00	0	0	0	0	0	$a_{11} \rightarrow \#000$
2	001	000	*	03	1	0	0	0	0	$a_{11} \cdot a_{1j} \rightarrow \#211$
3	191	211	191	02	0	0	0	0	0	} $a'_{ij}$ の punch out
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
58	029	000	*	03	1	0	0	0	0	
59	191	211	191	02	0	0	0	0	0	
60	030	000	*	03	1	0	0	0	0	
61	220	211	191	02	0	0	0	0	0	
62	002	001	*	15	0	0	0	0	0	call R-2 or R-1 by SNO

4 step より 58 step 迄は, odd-step はそのまま, 他は A-address のみを 1 づつあげる.

註 32 元のレオンチエフ行列の逆転は, この routine を 32 元専用直して用いた.  
 穿孔機 191 より  $A^{-1}$  が punch out され,  $\det |A|$  は print out される.

[附表 3] 逆行列計算用ルーティン (内部記憶のみによる)

計算式										INPUT 初期条件
R-1 (endless)										$n^* \rightarrow \#119$ $n^{2*} \rightarrow \#118$ $n \rightarrow \#117$ $1 \rightarrow \#184$ 原行列を #1~# $n^2$ に入れる。
ステップ	A	B	C	O			D			備考
				1	2	3	1	2	3	
1	001	*	183	00	0	0	0	0	0	$a_{11} \rightarrow 183$
2	183	184	184	03	0	0	0	0	0	$\det  A  \rightarrow \#184$
3	*	*	*	26	1	0	0	1	1	$a_{11}, \det  A $ の print
4	221	183	181	04	0	0	0	0	0	$a_{11}^{-1} \rightarrow \#181$
5	001	*	*	13	1	0	1	0	0	1 $\rightarrow$ SA-A
6	118	*	197	00	0	0	0	0	0	$n^2 \rightarrow$ SA-C
7	119	221	188	09	0	1	0	0	0	$(n-1) \rightarrow$ SNO
8	085	*	*	14	0	0	0	0	0	call R-85
9	119	*	195	00	0	0	0	0	0	$n \rightarrow$ SA-A
10	219	*	197	00	0	0	0	0	0	0 $\rightarrow$ SA-C
11	118	*	196	00	0	0	0	0	0	$n^2 \rightarrow$ SA-B
12	119	221	189	09	0	1	0	0	0	$(n-1) \rightarrow$ SNI
13	005	*	*	14	0	0	0	0	0	call R-5
14	118	*	195	00	0	0	0	0	0	$n^2 \rightarrow$ SA-A
15	118	119	197	09	0	1	0	0	0	$n(n-1) \rightarrow$ SNO
16	119	221	188	09	0	1	0	0	0	$(n-1) \rightarrow$ SNO
17	006	*	*	14	0	0	0	0	0	call R-6
18	181	*	000	00	0	0	0	0	1	$a_{nn} \rightarrow \#n$
19	117	221	117	11	0	0	0	0	0	計算回数の判別
20	001	066	*	17	0	0	0	0	0	判別結果に依り R-1を続けて読め。 終了したら R-66 を呼べ。

\*  $n^*$  は非正規化の  $n$  を示す。例えば  $10^*$  は  $0.0000010 \times 10^7$ 。 ( $n$  は行列の次数)

(output) 穿孔機 #191 より,  $A^{-1}$  のテープが punch out される。

カードルーティン

ステップ	A	B	C	O			D			備考
				1	2	3	1	2	3	
R-85										
1	000	181	000	03	0	0	1	0	1	$a_{1j}'$ の計算
2	085	001	*	15	0	0	0	0	0	call R-85 or R-1 by SNO
R-5										
1	119	221	188	09	0	1	0	0	0	$(n-1) \rightarrow$ SNO
2	000	*	182	00	0	0	1	0	0	$a_{11} \rightarrow \#182$
3	065	*	*	14	0	0	0	0	0	call R-65
R-65										
1	182	000	*	03	1	0	0	1	0	} $a_{ij}'$ の計算
2	000	211	000	02	0	0	1	0	1	
3	065	075	*	15	0	0	0	0	0	

## R-75

1	182	181	*	03	1	0	0	0	0	
2	220	211	000	02	0	0	0	0	1	
3	118	*	196	00	0	0	0	0	0	$n^2 \rightarrow SA-B$
4	005	001	*	16	0	0	0	0	0	call R-5 or R-1 by RNI

## R-6

1	000	*	000	00	0	0	1	0	1	ベクトルの転送
2	006	001	*	15	0	0	0	0	0	call R-6 or R-1 by SNO

## R-66

1	76	*	*	13	1	2	1	0	0	R-76 $\rightarrow$ STO
2	219	*	195	00	0	0	0	0	0	$o \rightarrow SA-A$
3	119	*	188	00	0	0	0	0	0	$n \rightarrow SNO$
4	090	*	*	14	0	2	0	0	0	逆行列 $A^{-1}$ の punch out 終つたら R-76 を呼べ。

## R-76

1	*	*	20	25	0	1	0	0	0	穿孔機 20 step jump
2	000	*	*	14	0	0	0	0	0	操作台を呼べ (計算終了)