

UDC 512.83:
681.3.06

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-348

1 次方程式系の解法 II

——一般の正方行列を係数行列とする場合——

福田正大・小松増美・末松俊二

1978年4月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

目 次

1. 緒 言	1
2. 行列の条件数とテスト行列の作成	2
3. 稠密行列の LU 分解法と方程式系の解法	4
3.1 方法の基礎	4
3.2 計算方法とサブルーチンの説明	5
3.3 数値的議論	6
3.4 サブルーチンの使用方法	7
3.5 数値テスト及び結果	12
4. 帯状行列の LU 分解法と方程式系の解法	20
4.1 方法の基礎	20
4.2 計算方法とサブルーチンの説明	21
4.3 数値的議論	23
4.4 サブルーチンの使用方法	24
4.5 数値テスト及び結果	30
5. 結 語	38

1 次方程式系の解法 II*

——一般の正方行列を係数行列とする場合——

福田正大**・小松増美**・末松俊二**

概 要

本報告は、最も基本的と思われる1次方程式系

$$Ax = b$$

を、 A が実又は複素稠密行列であるとき及び実帯状行列であるときの2通りの場合に分けて取り扱ったものである。解法としては直接法であるLU-分解法を使用した。さらに、多くのテスト結果によってこのような簡単な問題を解くのに種々の問題点があることを、特にライブラリー使用者に向けて指摘した。なお、実帯状行列の場合についてのみ inverse iteration 法による固有ベクトルの計算法をも掲載した。(ここで作成したLU-分解法のサブルーチンを使用しているためである)

1 章 結 言

本報告は、「1次方程式系の解法I」¹⁾にも述べてあるように、科学技術計算用サブルーチンライブラリーの標準化ならびに充実化という目標に沿ったものの1つである。このような研究は、アメリカでは連邦標準局などで行なわれているが、わが国では軽視される傾向にある内容のものである。しかしながら、基本的な各種解法や各種計算法の性質、適用範囲、精度、速度等といったものを明確にし、又系統的なテストを行ない各種情報を得るといったことは必要な、とりわけ計算センターにあっては大事な研究の一分野である。

今回報告するものは、1次方程式系としては最も基本的と思われる、 n 次正則行列 A を係数行列とする方程式系

$$Ax = b$$

を解くものである。解法としては直接法である、 A を下三角行列 L と上三角行列 U とによって

$$A = LU$$

と分解する、LU分解法を用いる。サブルーチンの種類としては、 A が実及び複素稠密行列である場合と実帯状行列の場合の3種類がある。

さらに、帯状行列についてのみ、近似固有値を与えて精度のよい固有ベクトルを求める inverse iteration 法

のサブルーチンがある。

なお、「1次方程式系の解法I」から次の2点でサブルーチンの外部仕様を変更した。

- (1) 1次元のワーキングエリアも引き数に加える。こうすることによって引き数の並びは若干煩雑になるが、その代わりに無駄なエリアを取らなくてすむ。
- (2) 整合配列の引き数を各2次元アレイごとにつけるのではなく、行列として同じ行数をもつものについては同じ整合配列引き数を用いた。

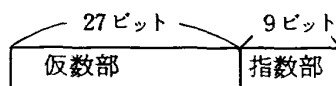
(例) A が $m \times n$ 、 B が $m \times r$ 、 X が $n \times r$ という行列のとき、 A と B には同じ整合配列引き数 KA が、 X には異なる引き数 KX が使用されている。従って、サブルーチン内の dimension 宣言は

$$\text{DIMENSION } A(KA, N), B(KA, R), \\ X(KX, R)$$

となっている。

内部的な変更としては、前記文献1)では行列式の値を指数部と仮数部とに分けるために、アセンブラ言語で書いたサブルーチンをコールしたが、以下のようにすればフォートランでも書けるので、そのような所は全てを書き直した。

単精度の場合を例にとると、230-75の浮動小数点数は、



となっており、かつ指数部は実際の値に $256=2^8$ を加

* 昭和53年2月13日 受付

** 計算センター

えたものになっている。それで、

```
EQUIVALENDE(D, ID)
DATA IX/0000000007770̄/, IY/7777777770000̄/
M=AND(ID, IX)-256
ID=AND(ID, IY)+256
```

とすれば、 M には D の指数部の実際の値がストアされ、新しい D の絶対値は $1/2$ と 1 の間になる。即ち、古い D の値 D_0 は、新しい D の値 D_N と M とによって

$$D_0 = D_N \times 2^M \quad 1/2 \leq |D_N| \leq 1$$

と書ける。

2章 行列の条件数とテスト行列の作成

2.1 条件数

行列 $A = (a_{ij})$ の逆行列を計算する問題に対する「条件数」なるものがいくつか提案されており、それは計算に関する行列の何らかの性質を表わしているものと思われる。その一般的と思われる形は、適当な行列のノルムに対する

$$P(A) = \|A\| \cdot \|A^{-1}\|$$

であり、TM-277¹⁾では行列ノルムとしてスペクトルノルムを用いた場合の数值を誤差の評価に用いた。これ以外にも、ノイマンとゴールドスタインは

$$K(A) = \max_i |\lambda_i| / \min_i |\lambda_i|$$

を提案しており、又チューリングは

$$M(A) = n \max_{i,j} |a_{ij}| \max_{i,j} |a_{ij}|$$

$$N(A) = 1/n \|A\|_E \|A^{-1}\|_E$$

の二つを提案している。ここで、 n は行列 A の次数、 λ_i はその固有値であり

$$A^{-1} = (a_{ij})$$

$$\|A\|_E = \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2}$$

である。これらの値の間には

$$1/n^2 M(A) \leq N(A) \leq M(A)$$

$$K(A) \leq nM(A)$$

という関係があり、さらに A が対称であれば

$$M(A) \leq nK(A)$$

が成り立つ。

2.2 テスト行列

いくつかのテスト行列からそれらを複合した次数の高いテスト行列を作る有効な方法の一つにクロネッカー積を用いるものがある。 n 次行列 $A = (a_{ij})$ と m 次行列 B

とより得られる mn 次行列

$$\begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nn}B \end{pmatrix}$$

を A と B のクロネッカー積と呼び、記号 $A \otimes B$ で表わす。そうすると

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

となる。これは、帯巾の小さい帯状行列から帯巾の大きい行列を作るのによい方法である。又、固有値問題のテスト行列作成に応用の広いものである。次にいくつかのテスト行列の構成法を述べる。以下の行列ではその逆行列が理論的に求まる利点がある。

(1) プレナーによるもの

全要素が1である $n \times m$ 行列を J_{nm} で表わす。 I_n は n 次単位行列である。

$$A = \begin{pmatrix} a_1 I_{n_1} + b_{11} J_{n_1 n_1}, & b_{12} J_{n_1 n_2}, & \dots, & b_{1m} J_{n_1 n_m} \\ b_{21} J_{n_2 n_1}, & a_2 I_{n_2} + b_{22} J_{n_2 n_2}, & \dots, & b_{2m} J_{n_2 n_m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} J_{n_m n_1}, & b_{m2} J_{n_m n_2}, & \dots, & a_m I_{n_m} + b_{mm} J_{n_m n_m} \end{pmatrix}$$

は、ブロック対角行列

$$\Gamma = \begin{pmatrix} A_1 & & 0 \\ & A_2 & \\ 0 & & \ddots \\ & & & A_{m+1} \end{pmatrix}$$

に相似で、各 A_i は

$$A_i = a_i I_{n_i - 1}$$

$$A_{m+1} = \begin{pmatrix} a_1 + b_{11} n_1, & b_{12} n_2, & \dots, & b_{1m} n_m \\ b_{21} n_1, & a_2 + b_{22} n_2, & \dots, & b_{2m} n_m \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} n_1, & b_{m2} n_2, & \dots, & a_m + b_{mm} n_m \end{pmatrix}$$

で与えられる。これを $m=2$ の場合に適用し、 n_1 を n 、 n_2 を m と改めて書くと

$$B = \begin{pmatrix} a I_n + b J_{nn}, & c J_{nm} \\ d J_{mn}, & e I_m + f J_{mm} \end{pmatrix}$$

は、

$$\Gamma = \begin{pmatrix} a I_{n-1} & 0 & 0 \\ 0 & b I_{m-1} & 0 \\ 0 & 0 & B_3 \end{pmatrix}$$

に相似で、

$$B_3 = \begin{pmatrix} a + b n & c m \\ d n & e + f m \end{pmatrix}$$

となる。このとき B の逆行列は

$$B^{-1} = \begin{pmatrix} a^{-1} I_n + b' J_{nn}, & c' J_{nm} \\ d' J_{mn}, & e^{-1} I_m + f' J_{mm} \end{pmatrix}$$

と書かれ

$$\begin{aligned} b' &= (e+fm-a^{-1}\sigma)/\sigma n \\ c' &= -c/\sigma \\ d' &= -d/\sigma \\ f' &= (a+bn-e^{-1}\sigma)/\sigma m \\ \sigma &= (a+bn)(e+fm)-cdnm \end{aligned}$$

となる。

(2) ニューベリ-によるもの
次の形の行列を考える。

$$A = \begin{pmatrix} S & R \\ C & D \end{pmatrix}$$

ここで、 S はスカラー、 R は行ベクトル (r_2, \dots, r_n) 、 C は列ベクトル $(c_2, \dots, c_n)^T$ 、 D は対角要素が d_2, \dots, d_n の対角行列である。そうすると、 A^{-1} を A と同じ形の部分行列に分解して、

$$A^{-1} = \begin{pmatrix} S' & R' \\ C' & M' \end{pmatrix}$$

と表わせば、 M' は一般的にはもはや対角行列ではなくなり、 $R' = (r'_2, \dots, r'_n)$ 、 $C' = (c'_2, \dots, c'_n)^T$ 、 $M' = (m'_{ij})$ ($2 \leq i, j \leq n$) とすれば、各値は

$$\begin{aligned} S' &= (S - \sum_{i=2}^n r_i c_i / d_i)^{-1} \\ c_i &= -S' c_i / d_i \\ r_i &= -S' r_i / d_i \quad 2 \leq i, j \leq n \\ m'_{ij} &= (\delta_{ij} - c_i r'_j) / d_i \end{aligned}$$

となる。 δ_{ij} はクロネッカーの δ である。

(3) オルテガによるもの

U, V を n 次の列ベクトル、 I を n 次単位行列として、
 $C = I + UV^*$

とする。そうすると C は正則でその逆行列は

$$C^{-1} = I - (1 + V^*U)^{-1} UV^*$$

で与えられる。 R を正則な n 次行列として、 C による R の相似変換を A とすれば、

$$A = CRC^{-1} = R + UV^*R - \alpha RUV^* - \alpha(V^*RU)UV^*$$

と表わされる。ここで $\alpha = (1 + V^*U)^{-1}$ 。逆行列は

$$A^{-1} = CR^{-1}C^{-1} = R^{-1} + UV^*R^{-1} - \alpha R^{-1}UV^* - \alpha(V^*R^{-1}U)UV^*$$

となる。この方法は適用範囲が広く、 U, V, R を適当に選べば、 A, A^{-1} を計算機内で正確に計算することが可能となる。例えば、 U と V を直交するようを選び、 R として対角要素が d_1, \dots, d_n の対角行列 D を採れば、 $\sigma = V^*DU$ として

$$A = D + UV^*D - DUV^* - \sigma UV^*$$

と表わせる。さらにこの例で、 n を偶数、 $U = (c, \dots, c,$

$c)^T$, $V = (1, \dots, 1, -1, \dots, -1)^T$ とすれば

$$a_{ij} = \begin{cases} d_i \delta_{ij} - c(d_i - d_j + \sigma) & 1 \leq j \leq n/2 \\ d_i \delta_{ij} + c(d_i - d_j + \sigma) & n/2 + 1 \leq j \leq n \end{cases}$$

となる。

以上いくつかのテスト行列構成法を述べたが、これら以外にももう少し自由度の少ない一概ね、パラメータとしては行列の次数だけ - ものも種々提案されている。以下に、実際に使用したテスト行列について述べる。

単純な稠密行列として、次数が小さくても条件数の大きい行列として代表的なロトキン行列 L_N 、オルテガ行列から4種類 $O_N^{(i,P)}$ と $O_N^{(i,D)}$ ($i=1, 2$)及び次に述べる帯状行列 B_N^2 の逆行列 T_N^2 の6種類。又、帯状行列としては、[1]で用いた B_N とそれを自乗した B_N^2 及びクロネッカー積 $B_N \otimes O_M^{(1,D)}$ 、 $B_N \otimes O_M^{(2,D)}$ の4種類を用いた。それぞれの行列の具体的な形は次の通りである。

(i) ロトキン行列 L_N

$$(L_N)_{ij} = \begin{cases} 1 & i=1, 1 \leq j \leq N \\ 1/(i+j-1) & 2 \leq i \leq N, 1 \leq j \leq N \end{cases}$$

$$(L_N)^{-1}_{ij} = \begin{cases} (-1)^{n-i} c_{i-1} \times c_i & 1 \leq i \leq N, j=1 \\ (-1)^{i-j+1} c_{i+j-1} \times c_{i+j-2} \times c_{j-1} \times c_{i+j-1} \times c_{i+j-1} \times c_i & 1 \leq i \leq N, 2 \leq j \leq N \end{cases}$$

(ii) オルテガ行列

(3)で述べた行列 R として対角行列を採りその対角要素として

$$(P)_i = i - p_{2m} + 1 \quad p_{2m} \leq i < p_{2m+2} \quad n=0, 1, \dots \\ (D)_i = (-1)^i i$$

の2種類を採る。ここで p_i は i 番目の素数で、 $p_0=1$ とする。又ベクトル U, V として

$$(U)_i = c, \quad (V)_i = 1 \quad 1 \leq i \leq N/2 \\ -1 \quad N/2 + 1 \leq i \leq N$$

$$(V)_i = 1/\sqrt{N}, \quad U = -2V$$

の2組を採り、前者を1、後者を2と番号づける。このように U, V, R を与えて(3)の定義に従って計算したのが $O_N^{(i,P)}$ と $O_N^{(i,D)}$ ($i=1, 2$)である。

(iii) B_N

$$(B_N)_{ij} = \begin{cases} 2 & 1 \leq i=j \leq N \\ -1 & |i-j|=1 \quad 1 \leq i, j \leq N \\ 0 & |i-j| \geq 2 \quad 1 \leq i, j \leq N \end{cases}$$

$$(B_N^{-1})_{ij} = \begin{cases} i(N-j+1)/(N+1) & i \leq j \\ = (B_N^{-1})_{ji} & i > j \end{cases}$$

(iv) B_N^2

$$(B_N^2)_{ij} = \begin{cases} 5 & i=j=1, i=j=N \\ 6 & 1 < i=j < N \\ -4 & |i-j|=1 \\ 1 & |i-j|=2 \\ 0 & |i-j| > 2 \end{cases} \quad 1 \leq i, j \leq N$$

$$((B_N^2)^{-1})_{ij} = \begin{cases} i_j M(2N+1) - i_j(i-1)(3N+2-i) \\ \quad - j(j^2-1)(N+1-i) \end{cases} / 6(N+1) \quad i \geq j \\ = ((B_N^2)^T)_{ji} \quad i < j$$

これらの行列を実際に使うときには、特に断わらない限り、適当に整数倍して要素が整数になるようにした。又、テストは実稠密行列を係数とする場合を重点として行った。又、テスト計算はすべて、FACOM 230-75 を使用し、フォートランH, V02, L09, OPT2 のもとに行った。

なお、本報告で用意したサブルーチンは、方程式系を解くものであって逆行列を直接求めるものではないが、テストは主として逆行列を求める計算について行った。結果の整理には、[1] と同じく、誤差行列として絶対誤差行列と相対誤差行列を採り、その各々の絶対値の最大値 m と自乗平均 f の二つの値を用いた。精度と条件数の関係を見るためには、精度は相対誤差の m と f を、条件数は計算の容易な N -条件数と M -条件数を用いた。 K は計算機に依存する値で、仮数部桁数を t としたとき、 $K=2^t$ である。なお、真値零に対応する相対誤差は絶対誤差に等しいとした。又、各種数値の表記で、『1.24-4』は 1.24×10^{-4} を表わす。

又、サブルーチンの使用方法の説明で、“アレイの大きさは KA ”，“ $N \times M$ 2次元アレイ” という用語を用いているが、この意味は次の通りである。アレイ A が1次元の場合、その大きさが KA であるとは、 A の DIMENSION が KA かもしくはそれより大きいことである。即ち

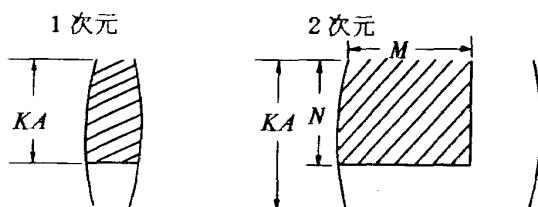
$$\text{DIMENSION } A(KA)$$

$$\text{DIMENSION } A(L) \quad L \geq KA$$

である。 A が2次元で、“大きさが KA の $N \times M$ 2次元アレイ” というのは、 A の DIMENSION 宣言が、

$$\text{DIMENSION } A(KA, L) \quad KA \geq N, L \geq M$$

となっていることである。図示すれば



となる。(斜線部分が実際に使用される領域である)

3章 密行列の LU 分解と方程式系の解法

3.1 方法の基礎

行列 A が正則行列で、そのすべての主小行列式の値が零でなければ、下三角行列 L と上三角行列 U とによって A を

$$A = LU$$

という形に分解できる。この分解は零でない対角要素の値を除いて一意である。通常のアプローチは、 L の対角要素を1にとるドットル分解か、 U の対角要素を1にとるクラウト分解かのどちらかである。ドットル分解は、単一除算手順に従った場合のガウス消去法に一致する。

ここでは、 U の対角要素を1にとるクラウト分解を採用する。このとき、 L と U の要素は L の第 r 列について U の第 r 行という順に、式

$$\sum_{k=1}^{r-1} l_{ik} U_{kr} + l_{ir} = a_{ir} \quad i = r, \dots, n$$

$$\sum_{k=1}^{r-1} l_{rk} U_{ki} + l_{rr} U_{ri} = a_{ri} \quad i = r+1, \dots, n$$

に従って求まる。この手続きでは、或る r に対して $l_{rr} = 0$ となったとき、それ以上の計算ができなくなるが、 A のすべての主小行列式の値が零でなければこの事態は生じず、上式に従って L と U の全要素が一意に定まる。 A が正則であってもどれか一つの主小行列式の値が零であれば $l_{rr} = 0$ となり計算は不可能になる。又 l_{rr} が零ではなくとも非常に小さい値になれば、計算に不必要な不安定性をもたらすことになる。このことを避けるために partial pivoting という操作を行なう。この効果は元の行列 A の行を入れ換えて得られる行列 \tilde{A} を分解することにある。即ち置換行列を P としたとき、

$$PA = \tilde{A} = \tilde{L}\tilde{U} = (\tilde{L}D)(D^{-1}\tilde{U}) = LU \quad (3.1)$$

となる。ここで $\tilde{L}\tilde{U}$ は \tilde{A} のドットル分解であり、 D は $D^{-1}\tilde{U}$ の対角成分が1になるような対角行列、従って LU は \tilde{A} のクラウト分解である。このとき、

$$|\tilde{l}_{ij}| \leq 1 \quad j < i$$

という関係は成り立つが、 $|\tilde{l}_{ij}| \leq 1$ とか $|U_{ji}| \leq 1$ ($j < i$) という関係は必ずしも満足されない。が

$$|l_{ij}| \leq |l_{jj}| \quad j < i$$

ではある。そうではあるが、クラウト分解もドットル分解と同じ数値的安定性を有していることが誤差解析で示されている。

この分解を行くと、 A の行列式 $\det(A)$ の値が

$$\det(A) = \prod_{i=1}^n l_{ii}$$

から求まる。こうして L と U が得られたならば、1次方程式系

$$Ax = b$$

の解 x が

$$Ly = Pb, \quad Ux = y$$

を解くことによって得られる。 $Ly = Pb$ は通常の前進過程であり、 $Ux = y$ は逆進過程である。こうして得られた解を精密化するには

$$\begin{aligned} x^{(0)} &= 0, \quad r^{(k)} = b - Ax^{(k)} \\ LUy^{(k)} &= Pr^{(k)} \quad x^{(k+1)} = x^{(k)} + y^{(k)} \end{aligned} \quad (3.2)$$

という逐次計算を行えばよい。この過程は、 A の条件数があまり大きくなくかつ $b - Ax^{(k)}$ の計算を倍精度で行うならば、必要な精度でもって収束することが示されている。

3.2 計算方法とサブルーチンの説明

実際のアルゴリズムは次の手順で行う。

$$(1) \quad l_{ir} = a_{ir} - \sum_{k=1}^{r-1} l_{ik} U_{kr} \text{ を計算して } a_{ir} \text{ へ over}$$

write する。 $i = r, \dots, n$

$$(2) \quad |l_{i_r, r}| = \max_{i \geq r} |l_{ir}| \text{ となる最小の } i_r \text{ を探して } i_r$$

をストアし、 i_r 行と r 行とを入れ換える。

$$(3) \quad U_{ri} = (a_{ri} - \sum_{k=1}^{r-1} l_{rk} U_{ki}) / l_{rr} \text{ を計算して } a_{ri} \text{ へ}$$

overwrite する。 $i = r+1, \dots, n$

途中段階を $n=5, r=3$ として図示すれば

$$\begin{pmatrix} l_{11} & U_{12} & U_{13} & U_{14} & U_{15} \\ l_{21} & l_{22} & U_{23} & U_{24} & U_{25} \\ l_{31} & l_{32} & a_{33} & a_{34} & a_{35} \\ l_{41} & l_{42} & a_{43} & a_{44} & a_{45} \\ l_{51} & l_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \end{pmatrix}$$

となる。この手続きを A に n 回施すと、 \tilde{A} の LU 分解が完成され、 L と U の値は A がストアされていた場所にストアされ、 i_r の値が A を \tilde{A} に変換する置換行列 P の詳細を教えてくれる。置換行列 $P_r = (p_{ij})$ を

$$\begin{cases} p_{ii} = 1 & i \neq r, \quad i_r : p_{i, i_r} = p_{i_r, i} = 1 \\ p_{ij} = 0 & \text{その他} \end{cases}$$

で定義すると、(3.1)-式の P は

$$P = P_n P_{n-1} \dots P_1$$

で与えられる。

A の各列を2の巾乗倍しても誤差は生じないし、又選ばれたピボット要素も変化しない。しかし行を何倍かすることはピボット要素の選択に影響を及ぼす。例えば、行列

$$A = \begin{pmatrix} \varepsilon & b \\ a & c \end{pmatrix}$$

をとる。ここで、 ε は a, b, c に比べて小さいとする。このとき、このままだと第1行と第2行の入れ換えが行われて不安定性が緩和される。が、第1行を適当倍すれば行交換はなくなり、以前より不安定な計算を行うことになる。ピボティングを行うアルゴリズムには多かれ少なかれこのようなスケールングの問題が含まれており、取り扱いがやっかいである。ここでは、 A の各行の長さを1にするようなスケールング因子 S_i を採用する。即ち

$$S_i = \left(\sqrt{\sum_{j=1}^n a_{ij}^2} \right)^{-1}$$

但し、この因子を実際に行列 A に掛けると、不必要な丸め誤差を最初からもたらすことになり、又方程式系 $Ax = b$ を解くときに、 b の各要素にも同じ因子を乗じる必要が起きる。それで、 A はそのままにしておいて(2)の段階で i_r を探す際に

$$|S_{i_r} \cdot l_{i_r r}| = \max_{i \geq r} |S_i l_{ir}|$$

として行う。以上に述べたような分解を、実行列に対して行うものがRLGDTS、複素行列に対して行うものがCXGDTSである。

ここで作成したサブルーチンは、方程式の右辺が複数個ある場合 $Ax_i = b_i \quad i=1, \dots, m$ ($AX=B$ とも書く。 X, B は $n \times m$ 行列) にも1回の呼び出して適用できる。係数行列を分解した後係を解くの用に用いるのが、RLGSLS と CXGSLS である。

逐次過程

$$\begin{aligned} x_i^{(0)} &= 0, \quad r_i^{(k)} = b_i^{(k)} - Ax_i^{(k)} \\ LUy_i^{(k)} &= Pr_i^{(k)}, \quad x_i^{(k+1)} = x_i^{(k)} + y_i^{(k)} \end{aligned}$$

の収束判定は次のようにする。

$$\|y_i^{(k)}\|_{\infty} \leq 2 \times \text{EPS} \times \|x_i^{(k+1)}\|_{\infty} \quad \text{EPS} = 2^{-t+1}$$

t : 仮数部桁数 (3.3)

がすべての $i=1, \dots, m$ に対して成立したとき逐次過程は収束したと判定する。又、或る段階 $k > 1$ で

$$\begin{aligned} \max_i (\|y_i^{(k)}\|_{\infty} / \|x_i^{(k+1)}\|_{\infty}) > \\ \frac{1}{2} \max_i (\|y_i^{(k-1)}\|_{\infty} / \|x_i^{(k)}\|_{\infty}) \end{aligned} \quad (3.4)$$

が成り立ったとき、行列の条件数が甚だ大きいと判断し

て逐次計算を打ち切り、異常終了する。この方法で系を解くのがRLASLS と CXASLS である。

3.3 数値的議論

計算された行列 LU は次の関係を満足する。

$$LU = \tilde{A} + \tilde{E}, \quad \tilde{A} = PA$$

ここで、 \tilde{E} は誤差行列、 P は置換行列である。 \tilde{E} の上限は内積計算の性質と、途中段階での全行列に対する絶対値最大の要素の大きさに依存する数 c 等に関する。内積が倍精度で計算されたときには、

$$|\tilde{e}_{ij}| \leq c \cdot 2^{-t} \quad t: \text{仮数部桁数}$$

となり、 c が $\max_{i,j} |a_{ij}|$ よりはるかに大きいということは普通にはない。前進過程、逆進過程による誤差は分解

誤差に比べて小さいので、 $Ax = b$ の計算された解 \bar{x} は

$$\bar{x} = (A+E)^{-1}b$$

としてよい。ここで E は、 \tilde{A} の誤差行列 \tilde{E} に対応する A の誤差行列である。これより

$$\|\bar{x} - A^{-1}b\|_2 / \|A^{-1}b\|_2 \leq \|E\|_2 \cdot \|A^{-1}\|_2 / (1 - \|E\|_2 \|A^{-1}\|_2)$$

が得られる。ここで $\|E\|_2 \|A^{-1}\|_2 = \alpha < 1$ ならば

$$\|\bar{x} - A^{-1}b\|_2 / \|A^{-1}b\|_2 \leq \alpha / (1 - \alpha)$$

となる。同様に、倍精度の内積計算を用いて剰余を計算すれば、逐次過程の各近似解 $x^{(k)}$ は、

$$\|x^{(k+1)} - A^{-1}b\|_2 \leq \alpha / (1 - \alpha) \cdot \|x^{(k)} - A^{-1}b\|_2$$

を満足する。詳しくは [1], [2], [3] を参照。

サブルーチン I

(1) RLGDTs, RLGDTD

このサブルーチンは稠密な実行列 A を $A=LU$ とクラウト分解するものである。副産物として A の行列式の値を与える。結果の L と U は、 A がストアされていた場所にストアされるので、元の A の値は失われる。なお、 U の対角要素は 1 とわかっているのでストアしない。

呼び出し形式

RLGDTs($N, A, KA, D, MD, INT, W, ILL$)

インプット

N : 行列 A の次数, $N \geq 3$

整数型又は整数型変数名

A : $N \times N$ 行列 A の要素をストアする。

大きさ KA の 2 次元実数型アレイ。

配列の仕方は通常の行列表示と同じである。

図 3-1 を参照

KA : アレイ A の大きさを指定, $KA \geq N$

整数型又は整数型変数名

アウトプット

A : インプット行列 A をクラウト分解した結果の下三角行列 L が下三角部分に, 上三角行列 U が上三角部分にそれぞれストアされている。図 3-1 を参照。

D, MD : 行列 A の行列式の値を $D \times 2^{MD}$ の形で与える。

MD は整数型変数名, D は実数型変数名で $1/2 \leq |D| \leq 1$

INT : 大きさ N の 1 次元整数型アレイ。

行交換の情報を保存している。

W : 大きさ N の 1 次元実数型アレイ。

ワーキングエリアとして使用する。

ILL : 整数型変数名

= 0 正常に終了したとき。

= K 第 K 番目のピボット要素が 2^{-t+4} より小さくなったとき, $1 \leq K \leq N$

t : 仮数部桁数

= 30000 入力パラメータに誤りがあったとき。

エラー処理

$KA < N$ 又は $N < 3$ の場合には, ILL 値をセットしただけでリターンする。従って, このときにはアレイ A の内容は元のままである。

第 K 番目のピボット要素が 2^{-t+4} より小さくなった場

合には, 行列 A は非正則であると判断して ILL 値をセットし, それより先の計算は打ち切ってリターンする。従って, このときにはアレイ A の内容は変化している。

詳細説明

インプット及びアウトプットのアレイ A の配列の仕方は次のとおりである。行列 A, L, U の要素をそれぞれ a_{ij}, l_{ij}, u_{ij} とする。

$$\begin{array}{c} \text{インプットアレイ } A \\ \left(\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right) \end{array} \quad \begin{array}{c} \text{アウトプットアレイ } A \\ \left(\begin{array}{cccc} l_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ l_{21} & l_{22} & u_{23} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{array} \right) \end{array}$$

図 3-1

行交換の情報は次のような形で保存されている。行列 A に対し第 K 行と第 $INT(K)$ 行との交換を $K=1, \dots, N$ について行って得られる行列を \tilde{A} とする。(即ち, 互換の積 $(N, INT(N)) \cdots (1, INT(1))$ を σ としたとき, 行列 A の行に置換 σ を施すと得られる。) この \tilde{A} をクラウト分解した場合の L, U がここに得られるものである。(4 節参照)

なお, $INT(1)$ と $W(1)$ とを EQUIVALENCE で結んでも支障はなく, こうすると使用領域を N 語減らすことができる。

RLGDTD を用いるときには, A, D, W を倍精度実数型として指定する。

(2) RLGSLS, RLGS LD

このサブルーチンは, 実正方行列 A を係数行列とする方程式 $AX=B$ を解くもので, A を $A=LU$ とクラウト分解し (RLGDTs), ついで $LY=B, UX=Y$ を解いて X を求める (RLGSLS)。

前項のサブルーチン(1)からのアウトプット A, INT が本サブルーチンのインプット A, INT になるので, このサブルーチンを使用する前に必ず(1)のサブルーチンを使用しておく。もしくは, それと同等な手続に従って A, INT に図 3-1 のような情報をセットする。一度そのようにセットされれば, このサブルーチンを何度使用してもよい。

解 X は B がストアされていた場所にストアされるので, B の元の値は失われる。

呼び出し形式

RLGSLS (N, NR, A, KA, INT, B)

インプット

N, NR : N は行列 A の次数, NR は方程式の右辺の
数, $N \geq 2, NR \geq 1$

整数又は整数型変数名。

A : 係数行列 A をクラウト分解して得られる三角行列
 L と U の要素を図3-1のようにストアする。

RLGDTS からのアウトプット A でよい。

$N \times N$ 行列で大きさ KA の2次元実数型アレイ。

KA : アレイ A, B の大きさを指定, $KA \geq N$

整数又は整数型変数名。

INT : 係数行列 A をクラウト分解する際の行交換の情
報をストアする。

RLGDTS からのアウトプット INT でよい。

大きさ N の1次元整数型アレイ

B : 方程式の右辺である $N \times NR$ 行列 B の要素をスト
アする。配列の仕方は通常の行列表示と同じであ
る。

大きさ KA の2次元実数型アレイ。

アウトプット

B : 方程式 $AX=B$ の解 X がストアされている。

エラー処理

$KA < N$ 又は $N < 2$ 又は $NR < 1$ のときには何もせずに
そのままリターンする。 ILL 値のセットも行わない。

詳細説明

RLGSLD を用いるときには, A, B を倍精度実数型
として指定する。

(3) RLASLS, RLASLD

このサブルーチンは, (2)で述べた方程式 $AX=B$ を (3.
2) の逐次式に従って精密に解きたい場合に使用するも
のである。係数行列 A のクラウト分解には, (1)のサブ
ルーチンRLGDTS を用いる。又逐次式で修正行列を求め
る所では(2)のサブルーチンRLGSLS を用いている。

なお元の A, B の値は保存されている。

呼び出し形式

RLASLS ($N, NR, A, KA, C, INT, B, X, ITS,$
 W, ILL)

インプット

N, NR : N は行列 A の次数, NR は方程式の右辺の数,

$N \geq 2, NR \geq 1$

整数又は整数型変数名。

A : 係数行列 A の要素をストアする。

大きさ KA の2次元実数型アレイ。

KA : アレイ A, C, B, X, W の大きさを指定。

$KA \geq N$

整数又は整数型変数名。

C : 係数行列 A をクラウト分解して得られる三角行列
 L と U の要素を図3-1のようにストアする。

RLGDTS からのアウトプット A でよい。

$N \times N$ 行列で, 大きさ KA の2次元実数型アレイ。

INT : 係数行列 A をクラウト分解する際の行交換の情報
をストアする。

RLGDTS からのアウトプット INT でよい。

大きさ N の1次元整数型アレイ。

B : 方程式の右辺である $N \times NR$ 行列 B の要素をスト
アする。

大きさ KA の2次元実数型アレイ。

アウトプット

X : 方程式 $AX=B$ の解 X がストアされている。

$N \times NR$ 行列で, 大きさ KA の2次元実数型ア
レイ。

ITS : イタレーションの回数

整数型変数名。

W : $N \times NR$ 行列で, 大きさ KA の2次元実数型ア
レイ。ワーキングエリアとして使用する。

ILL : 整数型変数名

= 0 正常に終了したとき。

= 1 イタレーションが収束しなかったとき。

= 30000 入力パラメータに誤りがあったとき。

エラー処理

$KA < N$ 又は $N < 2$ 又は $NR < 1$ の場合には, ILL 値を
セットしただけで何もせずにリターンする。

イタレーション中に, 係数行列の条件数が大きくて逐
次過程は収束しない, と判断した場合には ILL に1をセ
ットし, それ以上の計算はせずにリターンする。判定基
準については3.2節を参照。このとき, X には最終近似
解がストアされているが, 有効数字の有無については何
ともいえない。

詳細説明

このサブルーチンを用いるには, それより前に係数行
列 A をクラウト分解する必要があり, 又元の係数行列の
値も必要とする。それで, 全体の使用方法の一例として

は次のようになる。まず、 C に A と同じ値をストアしてから、

```
CALL RLGDT5(N, C, KA, D, MD, INT, W1,  
            ILL)  
CALL RLASLS(N, NR, A, KA, C, INT, B,  
            X, ITS, W2, ILL)
```

とする。ここでワーキングエリアを減らしたければ、 $W1(1)$ と $W2(1,1)$ とをEQUIVALENCEで結べばよい。

RLASLDを用いるときには、 $A, C, B, X, W2$ を倍精度実数型として指定する。

サブルーチン II

(1) CXGDTS, CXGDTD

このサブルーチンは稠密な複素行列 A を $A=LU$ とクラウト分解するものである。副産物として A の行列式の値を与える。結果の L と U は、 A がストアされていた場所にストアされるので、元の A の値は失われる。なお、 U の対角要素は 1 とわかっているのでストアしない。演算は実部・虚部にわけて行ない、複素演算は使用しない。従って A を表わすために、実部と虚部の二つのアレイを必要とする。

呼び出し形式

CXGDTS ($N, AR, AI, KA, DR, DI, MD, INT, W, ILL$)

インプット

N : 行列 A の次数, $N \geq 3$

整数定数又は整数型変数名

AR, AI : N 次複素行列 A の実部を AR に虚部を AI にストアする。

共に大きさ KA の 2 次元実数型アレイ。

配列の仕方は通常の行列表示と同じである。

図 3-2 を参照。

KA : アレイ AR, AI の大きさを指定, $KA \geq N$

整数定数又は整数型変数名

アウトプット

AR, AI : インプットの複素行列 A をクラウト分解して得られる下三角行列 L と上三角行列 U の実部が AR に虚部が AI に、それぞれの下三角部分と上三角部分にストアされている。

図 3-2 を参照

DR, DI, MD : 複素行列 A の行列式の値を $(DR + iDI) \times 2^{MD}$ の形で与える。

MD は整数型変数名, $DR \cdot DI$ は実数型変数名で、

$1/2 \leq \max(|DR|, |DI|) \leq 1$ 又は $DR=DI=0$

INT : 大きさ N の 1 次元整数型アレイ。

行交換の情報を保存している。

W : 大きさ N の 1 次元実数型アレイ。

ワーキングエリアとして使用する。

ILL : 整数型変数名

= 0 正常に終了したとき

= K partial pivoting を行った行列の第 K 次主小行列式の値が零になったとき。

= 30000 入力パラメータに誤りがあったとき。

エラー処理

$KA < N$ 又は $N < 3$ の場合には、 ILL 値をセットしただけでリターンする。従って、このときにはアレイ AR, AI の内容は元のままである。

第 K 次主小行列式の値が零になった場合には、行列 A は非正則なので ILL 値をセットし、それより先の計算は打ち切ってリターンする。従って、このときにはアレイ AR, AI の内容は変化している。

詳細説明

インプット及びアウトプットのアレイ AR, AI の配列の仕方は次のとおりである。行列 A, L, U の要素をそれぞれ $b_{jk} + ic_{jk}, p_{jk} + iq_{jk}, u_{jk} + iw_{jk}$ とする。(b, c, p, q, u, w は実数)

$$\begin{array}{ccc} & \text{インプットアレイ} & \\ & \begin{array}{cc} AR & AI \end{array} & \\ & \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} & \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \end{array}$$

アウトプットアレイ

$$\begin{array}{ccc} & \begin{array}{cc} AR & AI \end{array} & \\ & \begin{pmatrix} p_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ p_{21} & p_{22} & u_{23} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{pmatrix} & \begin{pmatrix} q_{11} & w_{12} & w_{13} & \cdots & w_{1n} \\ q_{21} & q_{22} & w_{23} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ q_{n1} & q_{n2} & q_{n3} & \cdots & q_{nn} \end{pmatrix} \end{array}$$

図 3-2

行交換の情報は次のような形で保存されている。行列 A に対し、第 K 行と第 $INT(K)$ 行との交換を $K=1, \dots, N$ について行なって得られる行列を \tilde{A} とする。(即ち、互換の積 $(N, INT(N) \cdots (1, INT(1)))$ を σ としたとき、行列 A の行に置換 σ を施すと得られる。) この \tilde{A} をクラウト分解した場合の L, U がここに得られるものである。

なお、 $INT(1)$ と $W(1)$ とを EQUIVALENCE で結んでも支障はなく、こうすると使用領域を N 語減らすことができる。

CXGDTD を用いるときには、 AR, AI, DR, DI, W を倍精度実数型として指定する。

(2) CXGSLS, CXGSLD

このサブルーチンは、複素正方行列 A を係数行列とする方程式 $AX=B$ を解くもので、 A を $A=LU$ とクラウト分解し (CXGDTS)、ついで $LY=B$, $UX=Y$ を解いて X を求める (CXGSLS)。

前項のサブルーチン(1)からのアウトプット AR, AI, INT が本サブルーチンのインプット AR, AI, INT になるので、このサブルーチンを使用する前に必ず(1)のサブルーチンを使用しておく。もしくは、それと同等な手続に従って AR, AI, INT に図 3-2 のような情報をセットする。一度そのようにセットされれば、このサブルーチンを何度使用してもよい。

解 X は B がストアされていた場所にストアされるので、 B の元の値は失われる。

呼び出し形式

CXGSLS($N, NR, AR, AI, KA, INT, BR, BI$)

インプット

N, NR : N は行列 A の次数, NR は方程式の右辺の次数, $N \geq 2, NR \geq 1$

整数又は整数型変数名

AR, AI : 係数行列 A をクラウト分解して得られる三角行列 L と U の要素を図 3-2 のように実部と虚部に分けてストアする。

CXGDTS からアウトプット AR, AI でよい。

共に $N \times N$ 行列で、大きさ KA の 2 次元実数型アレイ。

KA : アレイ AR, AI, BR, BI の大きさを指定,

$KA \geq N$

整数又は整数型変数名。

INT : 係数行列 A をクラウト分解する際の行交換の情報をストアする。

CXGDTS からのアウトプット INT でよい。

大きさ N の 1 次元整数型アレイ。

BR, BI : 方程式の右辺である $N \times NR$ 複素行列 B の実部を BR に虚部を BI にストアする。配列の仕方は通常の行列表示と同じである。共に大きさ KA の 2 次元実数型アレイ。

アウトプット

BR, BI : 方程式 $AX=B$ の解 X の実部が BR に虚部が BI にストアされている。

エラー処理

$KA < N$ 又は $N < 2$ 又は $NR < 1$ の場合には何もせずにそのままリターンする。 ILL 値のセットも行なわない。

詳細説明

CXGSLD を用いるときには AR, AI, BR, BI を倍精度実数型として指定する。

(3) CXASLS, CXASLD

このサブルーチンは、(2)で述べた方程式 $AX=B$ を (3.2) の逐次式に従って精密に解きたい場合に使用するものである。係数行列 A のクラウト分解には、(1)のサブルーチン CXGDTS を用いる。又逐次式で修正行列を求める所では(2)のサブルーチン CXGSLS を用いている。

なお元の A, B の値は保存されている。

呼び出し形式

CXASLS($N, NR, AR, AI, KA, CR, CI, INT, BR, BI, XR, XI, ITS, W1, W2, ILL$)

インプット

N, NR : N は行列 A の次数, NR は方程式の右辺の次数, $N \geq 2, NR \geq 1$

整数又は整数型変数名

AR, AI : 係数行列 A の実部を AR に虚部を AI にストアする。

共に大きさ KA の 2 次元実数型アレイ

KA : アレイ $AR, AI, CR, CI, BR, BI, XR, XI, W1, W2$ の大きさを指定, $KA \geq N$

CR, CI : 係数行列 A をクラウト分解して得られる三角行列 L と U の要素を図 3-2 のように実部と虚部に分けてストアする。

CXGDTS からのアウトプット AR, AI でよい。

共に $N \times N$ 行列で、大きさ KA の 2 次元実数型アレイ。

INT : 係数行列 A をクラウト分解する際の行交換の情報をストアする。

CXGDTS からのアウトプット INT でよい。

大きさ N の 1 次元整数型アレイ。

BR, BI : 方程式の右辺である $N \times NR$ 複素行列 B の実部を BR に虚部を BI にストアする。

共に大きさ KA の 2 次元実数型アレイ。

アウトプット

XR, XI : 方程式 $AX=B$ の解 X の実部が XR に虚部が XI にストアされている。

共に $N \times N$ 行列で、大きさ KA の 2 次元実数型アレイ。

ITS : イタレーションの回数
整数型変数名。

$W1, W2$: 共に $N \times NR$ 行列で、大きさ KA の 2 次元実数型アレイ。

ワーキングエリアとして使用する。

ILL : 整数型変数名

- = 0 正常に終了したとき。
- = 1 イタレーションが収束しなかったとき。
- = 30000 入力パラメータに誤りがあったとき。

エラー処理

$KA < N$ 又は $N < 2$ 又は $NR < 1$ の場合には、 ILL 値をセットしただけで何もせずにリターンする。

イタレーション中に、係数行列の条件数が大きすぎて逐次過程は収束しない、と判断した場合には ILL に 1 をセットし、それ以上の計算はせずにリターンする。判定基準は 3-3 式及び 3-4 式を自乗した形で用いる。このとき XR, XI には最終近似解がストアされているが、有効数字の有無については何ともいえない。

詳細説明

このサブルーチンは RLSLS に対応するものであり、使用方法の例等については RLSLS の詳細説明を参照。

CXASLD を用いるときには、 $AR, AI, CR, CI, BR, BI, XR, XI, W1, W2$ を倍精度実数型として指定する。

3.4 数値テスト及び結果

まず始めに、RLGDTS と CXGDTS のアウトプット INT の内容を実際例で示すことにする。どちらも同じものなので、RLGDTS についてのみ記す。行列 A として

$$A = \begin{pmatrix} 1 & -2 & 3 & 1 \\ -2 & 1 & -2 & -1 \\ 3 & -2 & 1 & 5 \\ 1 & -1 & 5 & 3 \end{pmatrix}$$

を採り、これを RLGDTs でクラウト分解すると、得られるアウトプットの A 及び INT は

$$A = \begin{pmatrix} -2 & -1/2 & 1 & 1/2 \\ 1 & -3/2 & -4/3 & -1/3 \\ 1 & -1/2 & 10/3 & 7/10 \\ 3 & -1/2 & -8/3 & 26/5 \end{pmatrix}$$

$$INT = (2, 2, 4, 4)$$

である。これよりクラウト分解の L と U は

$$L = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 1 & -3/2 & 0 & 0 \\ 1 & -1/2 & 10/3 & 0 \\ 3 & -1/2 & -8/3 & 26/5 \end{pmatrix} \quad U = \begin{pmatrix} 1 & -1/2 & 1 & 1/2 \\ 0 & 1 & -4/3 & -1/3 \\ 0 & 0 & 1 & 7/10 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

で与えられる。これらを掛け合わせた行列 LU は、

$$LU = \begin{pmatrix} -2 & 1 & -2 & -1 \\ 1 & -2 & 3 & 1 \\ 1 & -1 & 5 & 3 \\ 3 & -2 & 1 & 5 \end{pmatrix}$$

となり、これは INT の情報より、元の行列 A の第 1 行と第 2 行を交換し、ついで第 3 行と第 4 行を交換して得られる行列と一致している。 $(INT(2)=2, INT(4)=4)$ なので、これは恒等置換となり実際には行交換は行われぬ。又、常に $INT(N)=N$ である。 N は行列 A の次数)

次に RLGDTs を用いた場合の行列式計算のチェックは、テスト行列

$$A_0 = \begin{pmatrix} -73 & 78 & 24 \\ 92 & 66 & 25 \\ -80 & 37 & 10 \end{pmatrix} \quad \det(A_0) = 1$$

を用いて行った。一般に、行列要素の攪乱が行列式に与える変化の大きさは、

$$d(\det A) = \sum_{i,j} \tilde{a}_{ij} da_{ij}$$

で与えられる。ここで \tilde{a}_{ij} は A の (i, j) -余因子である。 A_0 に対する \tilde{a}_{ij} から作られる行列を \tilde{A}_0 とすると、それは

$$\tilde{A}_0 = \begin{pmatrix} -265 & -2920 & 8684 \\ 108 & 1190 & -3539 \\ 366 & 4033 & -11994 \end{pmatrix} = (A_0^{-1})^T$$

である。これより行列 A_0 は、 a_{13}, a_{23}, a_{33} 等の攪乱に対して行列式の値が大きく変化する特質を有しているのがわかる。 a_{33} を $a_{33} + \alpha$ とした行列を A_1 、 a_{21} を $a_{21} + \alpha$ とした行列を A_2 、 a_{12} を $a_{12} + \alpha$ とした行列を A_3 とし、各 A_i の行列式を RLGDTs を用いて計算した結果は表 3-1 である。

これを見ると、各場合の絶対誤差即ち $|d(\det A_i)|$ の値はほぼ同じ大きさである。これは、計算途中に生じる行列要素の攪乱が同程度であるとすれば、 \tilde{A}_i の \tilde{A}_0 からの攪乱が高々 $92|\alpha|$ と小さいことから首肯できる結果である。

CXGDTS の行列計算のチェックには、前記 A_i に複素行列

$$C = \begin{pmatrix} 1+i & 0 & -i \\ -i & 1+i & i \\ 0 & i & 1+i \end{pmatrix} \quad \det C = -1+2i$$

を左から掛けて得られる行列 $C_i = CA_i$ を用いた。 C_i の行列式の値は

$$\det C_i = \det C \times \det A_i$$

で与えられる。計算した結果は表 3-2 である。なお絶対誤差、相対誤差それぞれの数値は、複素数としての絶

対値で与えてある。

このテスト行列 A_0 の行列式計算に対する安定性を見るために、 $\det(cA_0)/c^3$ の値を計算してグラフにした。(図3-3, $c = 1/(2n-1); n=1, 16$, 縦軸は相対誤差(=絶対誤差), 横軸は $2n-1$) このような現象は、行列そのものの性質が解法に影響を及ぼして生じたものである。これからでもわかるように、自分が解こうとしている行列が攪乱に対してどのような振舞をするかを知ることが、ライブラリーを用いるとき、さらに言えば或る問題を計算機で解く場合の基本的作業の一つであるという認識が必要である。

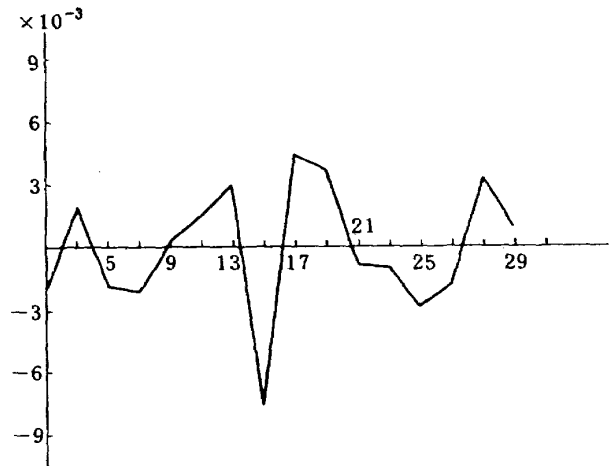


図 3-3

今度は、攪乱がLU-分解の精度に及ぼす影響を見るた

表 3-1

		A_0	A_1	A_2	A_3
$\alpha = 1/64$	絶対誤差	2.51 -3	4.01 -3	1.13 -3	2.51 -3
		3.27 -14	8.68 -14	1.14 -13	3.28 -14
	相対誤差	2.51 -3	2.15 -5	4.20 -4	5.62 -5
		3.27 -14	4.66 -16	4.24 -14	7.34 -16
det(A_i) の真値		1.0	$-186 \frac{13}{32}$	2.6875	-44.625
$\alpha = 1/128$	絶対誤差	2.51 -3	2.08 -3	1.02 -3	2.51 -3
		3.27 -14	7.42 -15	4.62 -14	3.29 -14
	相対誤差	2.51 -3	2.24 -5	5.51 -4	1.15 -4
		3.27 -14	8.01 -17	2.51 -14	1.51 -15
det(A_i) の真値		1.0	$-92 \frac{45}{64}$	$\frac{59}{32}$	$-21 \frac{13}{16}$

上段 RLGDTTS によるもの

下段 RLGDTD によるもの

表 3-2

		C_0	C_1	C_2	C_3
$\alpha = 1/64$	絶対誤差	1.29 -2	1.23 -2	8.33 -3	1.45 -2
		1.82 -13	2.20 -13	1.28 -13	2.27 -13
	相対誤差	5.79 -3	2.95 -5	1.39 -3	1.45 -4
		8.16 -14	5.28 -16	2.13 -14	2.28 -15
$\alpha = 1/128$	絶対誤差	1.29 -2	7.13 -3	7.35 -3	1.25 -2
		1.82 -13	1.83 -13	6.65 -14	2.45 -13
	相対誤差	5.79 -3	3.44 -5	1.78 -3	2.56 -4
		8.16 -14	8.82 -16	1.61 -14	5.01 -15

上段 CXGDTS によるもの

下段 CXGDTD によるもの

めに、定数倍して各要素が整数値になるようにしたロトキン行列と、定数倍していないロトキン行列とによる比較を表 3-3 に与える。この表で、 LU とあるのは分解して得られる三角行列 L と U を掛けてそれが元の A からどれだけ違っているか、又 L 、 U とあるのは 4 倍長で計算した L と U からどれだけ違っているかをそれぞれ相対誤差の最大値で表わしたものである。上段が定数倍したロトキン行列のもの、下段がそうでないロトキン行列のものである。これを見ると、積としての LU は元の行列 A と高々最後の 1 ビット程度の違いしかないことがわかる。しかし、 L と U そのものには相当の誤差が生じてお

り、これが方程式系を解く場合の誤差の大きな原因になる。(表 3-4, 5 を参照)

RLGSLS を用いて、2 章で述べた各種テスト行列の逆行列を計算した場合の誤差を表 3-4 ~ 3-13 に与える。表中の記号で m は最大値を f は自乗平均を表わし、 N_c と M_c はそれぞれ 2 章に書いてある条件数である。なお、単精度の L_N で $N=7, 8$ では、 ILL が零でないが、 $ILL=N$ のため分解して得られる行列 L と U は最後まで計算されているので逆行列の計算を行った。これらの表から、各々の場合に条件数のオーダー程の桁落ちが生じるといふ傾向がここでも見られるのがわかる。

表 3-3

		N	3	4	5	6	7	8	9	10
LU	単		0.0	2.72 ⁻⁸	1.82 ⁻⁸	2.20 ⁻⁸	2.17 ⁻⁸	2.17 ⁻⁸	——	——
			0.0	0.0	0.0	0.0	0.0	0.0	——	——
	倍		0.0	0.0	5.29 ⁻¹⁹	6.41 ⁻¹⁹	6.31 ⁻¹⁹	7.10 ⁻¹⁹	5.20 ⁻¹⁹	6.25 ⁻¹⁹
			0.0	0.0	0.0	0.0	4.34 ⁻¹⁹	4.34 ⁻¹⁹	4.34 ⁻¹⁹	4.34 ⁻¹⁹
L	単		1.94 ⁻⁷	1.45 ⁻⁶	2.65 ⁻⁵	5.14 ⁻⁴	8.43 ⁻³	8.31 ⁻²	——	——
			2.79 ⁻⁷	1.36 ⁻⁵	5.74 ⁻⁴	4.19 ⁻³	2.42 ⁻¹	5.87	——	——
	倍		6.07 ⁻¹⁸	4.12 ⁻¹⁷	3.10 ⁻¹⁶	3.17 ⁻¹⁵	3.39 ⁻¹³	1.68 ⁻¹¹	3.04 ⁻¹⁰	6.58 ⁻¹⁰
			4.88 ⁻¹⁸	4.55 ⁻¹⁶	8.59 ⁻¹⁵	3.10 ⁻¹⁴	5.66 ⁻¹²	2.18 ⁻¹⁰	4.04 ⁻⁹	5.50 ⁻⁸
U	単		1.99 ⁻⁸	5.79 ⁻⁸	4.22 ⁻⁷	1.42 ⁻⁵	3.79 ⁻⁵	3.86 ⁻⁴	——	——
			1.99 ⁻⁸	3.19 ⁻⁷	9.62 ⁻⁷	8.14 ⁻⁵	2.32 ⁻⁴	3.17 ⁻²	——	——
	倍		5.78 ⁻¹⁹	4.22 ⁻¹⁸	1.09 ⁻¹⁷	1.82 ⁻¹⁶	2.58 ⁻¹⁵	1.11 ⁻¹³	1.75 ⁻¹²	1.30 ⁻¹¹
			5.78 ⁻¹⁹	7.59 ⁻¹⁸	9.08 ⁻¹⁷	1.37 ⁻¹⁵	3.98 ⁻¹⁵	7.25 ⁻¹⁴	9.54 ⁻¹³	3.60 ⁻¹⁰

表 3-4

L_N ; Single

		N	3	4	5	6	7 [*]	8 ^{**}
絶対誤差	m		7.153 ⁻⁷	1.454 ⁻⁵	1.433 ⁻³	6.876 ⁻²	2.745	8.992 ⁺²
	f		3.284 ⁻⁷	6.179 ⁻⁶	5.212 ⁻⁴	2.227 ⁻²	8.348 ⁻¹	2.482 ⁺²
相対誤差	m		3.166 ⁻⁷	1.788 ⁻⁶	3.632 ⁻⁵	5.990 ⁻⁴	1.256 ⁻²	1.264 ⁻¹
	f		2.368 ⁻⁷	1.250 ⁻⁶	2.769 ⁻⁵	4.626 ⁻⁴	8.544 ⁻³	9.292 ⁻²
$\log K - \log N_c$			5.618	4.195	2.750	1.292	-1.761 ⁻¹	-1.652
$\log K - \log M_c$			5.094	3.589	1.999	4.499 ⁻¹	-1.080	-2.626
$-\log m$			6.499	5.748	4.440	3.223	1.901	8.983 ⁻¹
$-\log f$			6.626	5.903	4.558	3.335	2.068	1.032

* $ILL=7$ ** $ILL=8$

表 3-5

L_N ; Double

		N	5	6	7	8	9	10
絶対誤差	m		1.707 ⁻¹⁴	3.996 ⁻¹³	1.102 ⁻¹⁰	1.678 ⁻⁷	2.677 ⁻⁶	1.017 ⁻⁵
	f		6.303 ⁻¹⁵	1.306 ⁻¹³	3.293 ⁻¹¹	4.665 ⁻⁸	7.060 ⁻⁷	2.490 ⁻⁶
相対誤差	m		4.450 ⁻¹⁶	3.256 ⁻¹⁵	7.647 ⁻¹³	3.638 ⁻¹¹	5.802 ⁻¹⁰	1.386 ⁻⁹
	f		3.542 ⁻¹⁶	2.519 ⁻¹⁵	3.589 ⁻¹³	1.790 ⁻¹¹	3.198 ⁻¹⁰	7.631 ⁻¹⁰
$\log K - \log N_c$			13.29	11.83	10.36	8.884	7.403	5.916
$\log K - \log M_c$			12.54	10.99	9.456	7.91	6.383	4.836
$-\log m$			15.35	14.49	12.12	10.44	9.236	8.858
$-\log f$			15.45	14.60	12.45	10.75	9.495	9.117

表 3-6

$O_N^{(1,P)}$; Single

		N	10	50	100	200	300	400
絶対誤差	m		1.788 ⁻⁷	1.699 ⁻⁵	2.888 ⁻⁴	1.069 ⁻⁴	1.281 ⁻³	6.037 ⁻³
	f		6.429 ⁻⁸	1.209 ⁻⁵	2.352 ⁻⁴	8.266 ⁻⁵	1.112 ⁻³	5.636 ⁻³
相対誤差	m		1.192 ⁻⁷	5.946 ⁻⁶	6.371 ⁻⁵	2.794 ⁻⁵	1.851 ⁻⁴	4.255 ⁻⁴
	f		5.790 ⁻⁸	4.174 ⁻⁶	4.995 ⁻⁵	2.111 ⁻⁵	1.588 ⁻⁴	3.961 ⁻⁴
$\log K - \log N_c$			6.067	4.098	3.273	2.869	2.212	1.440
$\log K - \log M_c$			5.623	3.879	3.132	2.713	2.107	1.388
$-\log m$			6.924	5.226	4.196	4.554	3.733	3.371
$-\log f$			7.237	5.379	4.301	4.676	3.799	3.402

表 3-7

$O_N^{(1,D)}$; Single

		N	10	50	100	200	300	400
絶対誤差	m		2.384 ⁻⁷	1.079 ⁻⁵	3.368 ⁻⁶	4.840 ⁻⁵	5.913 ⁻⁵	6.372 ⁻⁵
	f		8.824 ⁻⁸	3.085 ⁻⁶	9.693 ⁻⁷	1.432 ⁻⁵	1.512 ⁻⁵	1.531 ⁻⁵
相対誤差	m		1.333 ⁻⁶	9.144 ⁻⁶	6.904 ⁻⁵	4.787 ⁻⁴	6.519 ⁻⁴	5.185 ⁻⁴
	f		2.518 ⁻⁷	3.872 ⁻⁶	2.097 ⁻⁶	2.114 ⁻⁵	2.223 ⁻⁵	2.221 ⁻⁵
$\log K - \log N_c$			5.647	4.415	4.064	3.467	3.116	2.867
$\log K - \log M_c$			4.966	3.601	3.190	2.585	2.232	1.982
$-\log m$			5.875	5.039	4.161	3.320	3.186	3.285
$-\log f$			6.599	5.412	5.678	4.675	4.653	4.653

表 3-8

 $O_N^{(2,P)}$; Single

		N	10	50	100	200	300	400
絶対誤差	m		9.313 ⁻¹⁰	2.910 ⁻¹¹	7.276 ⁻¹²	1.819 ⁻¹²	9.095 ⁻¹³	9.095 ⁻¹³
	f		1.411 ⁻¹⁰	1.828 ⁻¹²	2.627 ⁻¹³	3.739 ⁻¹⁴	1.201 ⁻¹⁴	7.240 ⁻¹⁵
相対誤差	m		5.953 ⁻⁸	2.431 ⁻⁷	5.428 ⁻⁷	1.818 ⁻⁶	1.818 ⁻⁶	3.135 ⁻⁶
	f		2.141 ⁻⁸	2.776 ⁻⁸	4.335 ⁻⁸	6.156 ⁻⁸	5.433 ⁻⁸	5.798 ⁻⁸
$\log K - \log N_c$			7.614	7.470	7.435	7.391	7.353	7.339
$\log K - \log M_c$			6.388	5.171	4.776	4.283	3.978	3.851
$-\log m$			7.225	6.614	6.265	5.740	5.740	5.504
$-\log f$			7.669	7.557	7.363	7.211	7.265	7.237

表 3-9

 $O_N^{(2,D)}$; Single

		N	10	50	100	200	300	400
絶対誤差	m		4.657 ⁻¹⁰	5.821 ⁻¹¹	1.819 ⁻¹²	9.095 ⁻¹³	9.095 ⁻¹³	4.547 ⁻¹³
	f		1.322 ⁻¹⁰	2.120 ⁻¹²	7.623 ⁻¹⁴	1.126 ⁻¹⁴	5.719 ⁻¹⁵	1.646 ⁻¹⁵
相対誤差	m		3.392 ⁻⁷	2.947 ⁻⁵	3.330 ⁻⁴	9.681 ⁻⁴	5.180 ⁻²	3.637 ⁻³
	f		6.601 ⁻⁸	1.111 ⁻⁶	4.426 ⁻⁶	8.139 ⁻⁶	1.748 ⁻⁴	1.262 ⁻⁵
$\log K - \log N_c$			7.439	7.104	6.955	6.806	6.718	6.656
$\log K - \log M_c$			6.238	4.500	3.862	3.242	2.884	2.631
$-\log m$			6.470	4.531	3.478	3.014	1.286	2.439
$-\log f$			7.180	5.954	5.354	5.089	3.758	4.899

表 3-10

 T_N^2 ; Single

		N	10	50	100	200	300	400
絶対誤差	m		3.679 ⁻⁸	1.105 ⁻⁶	6.439 ⁻⁶	1.160 ⁻⁴	/	/
	f		9.304 ⁻⁹	1.048 ⁻⁷	3.892 ⁻⁷	2.885 ⁻⁵	/	/
相対誤差	m		2.430 ⁻⁶	1.751 ⁻⁴	2.100 ⁻³	9.983 ⁻²	/	/
	f		3.207 ⁻⁷	5.531 ⁻⁶	2.757 ⁻⁵	4.653 ⁻³	/	/
$\log K - \log N_c$			5.239	2.915	1.876	8.306 ⁻¹	2.168 ⁻¹	-2.192 ⁻¹
$\log K - \log M_c$			4.606	1.908	7.169 ⁻¹	-4.807 ⁻¹	-1.183	-1.682
$-\log m$			5.615	3.757	2.678	1.001	/	/
$-\log f$			6.494	5.257	4.560	2.332	/	/

表 3-11

T_N^2 ; Double

		N	10	50	100	200	300	400
絶対誤差	m		1.520 -18	5.669 -17	1.583 -16	6.540 -16	1.886 -15	4.065 -15
	f		5.104 -19	3.698 -18	1.111 -17	3.681 -17	6.596 -17	1.088 -16
相対誤差	m		7.424 -17	1.193 -14	9.253 -15	2.085 -13	3.564 -13	2.139 -12
	f		1.356 -17	3.668 -16	1.733 -16	1.360 -15	2.171 -15	1.083 -14
$\log K - \log N_c$			15.78	13.45	12.41	11.37	10.75	10.32
$\log K - \log M_c$			15.14	12.44	11.25	10.06	9.353	8.854
$-\log m$			16.13	13.92	14.03	12.68	12.45	11.67
$-\log f$			16.87	15.44	15.76	14.87	14.66	13.97

表 3-12

$B_n \otimes O_{10}^{(1,D)}$; Single

$N=10n$

		N	50	100	200	300	400
絶対誤差	m		3.040 -6	1.347 -5	2.351 -4	4.520 -4	8.893 -4
	f		4.767 -7	2.047 -6	3.961 -5	7.031 -5	1.561 -4
相対誤差	m		1.037 -5	2.261 -5	4.522 -5	1.324 -4	2.072 -4
	f		1.119 -6	2.739 -6	1.577 -5	2.316 -5	4.238 -5
$\log K - \log N_c$			5.030	4.656	4.244	3.993	3.812
$\log K - \log M_c$			3.790	3.229	2.644	2.299	2.052
$-\log m$			4.984	4.646	4.345	3.878	3.684
$-\log f$			5.951	5.562	4.802	4.635	4.373

表 3-13

$B_n \otimes O_{10}^{(2,D)}$; Single

$N=10n$

		N	50	100	200	300	400
絶対誤差	m		1.863 -9	7.334 -9	4.913 -8	1.192 -7	2.161 -7
	f		2.056 -10	1.011 -9	5.213 -9	1.426 -8	2.515 -8
相対誤差	m		1.390 -6	4.780 -6	8.988 -6	2.246 -5	2.822 -5
	f		1.439 -7	5.981 -7	1.338 -6	2.393 -6	2.854 -6
$\log K - \log N_c$			6.821	6.447	6.035	5.785	5.604
$\log K - \log M_c$			5.062	4.501	3.917	3.571	3.324
$-\log m$			5.857	5.321	5.046	4.649	4.549
$-\log f$			6.842	6.223	5.873	5.621	5.544

又RLGDTS, RLGSLS 及びそれらの倍精度ルーチンで逆行列を計算するのに要する時間を表 3-14 に掲げる。RLGSLS を呼ぶ際のパラメーター NR は NC 等しくとってある。(RLGSLD の $N=400$ の場合のみ, $NR=50$ で計算して得られる値を 8 倍してある)。時間の後に書いてある数値は行交換の回数である。計算回数は N^3 に比例するので, 所要時間もやはり N^3 に比例しているのがわかる。なおここまでの表で, 斜線は $0 < ILL < N$ のため実測できていないことを示す。

逐次法で解の精度の向上を図った場合にはどうなるかを RLASLS を用いて, ロトキン行列をテスト行列にして逆行列を求めた場合の結果を表 3-15, 16 に示す。 d_{k+1} は (3.4)-式の左辺の値である。これからわかるように, 収束解の一つ手前で概ね収束解と同じものが得られているが, 収束判定のためにもう一度繰り返し部分を計算する必要がある。又単精度の場合の $N=7, 8$ によく現われているが, 一回の繰り返しごとに達成される精度の向上 ($-\log^{mk}/m_{k-1}, -\log^{fk}/f_{k-1}, -\log d_k/d_{k-1}$ 等の値, $N=7$ で 2.064, $N=8$ で 1.030) は, 繰り返しの終りの方を除いてほぼ一定しており, それは逐次法を用いない場合に得られる精度(表 3-4 参照)と一致している。このことは, 近似解に対する残差ベクトルの計算での誤差が無視できれば, 修正ベクトルの計算による桁落ちだけが精度の向上に影響するので, 妥当な結果である。なお, このテスト行列では最終の収束解の誤差が零になっているが, 収束すれば必ずそうであるという保証はない。そのことを示すために, 同じロトキン行列を定数倍することなく用いた場合の収束解の結果を表 3-17

に示す。この場合の収束の様子を見てみると 1 回の繰り返しごとに殆んど精度の向上が行われず, 修正ベクトルが相対的に零に近いまま収束している。これは, 定数倍していないロトキン行列の逆行列の要素はその絶対値が非常に大きく, ロトキン行列の要素を計算機内部で表現するとき生じる最後の 1 ビットの誤差と相まって, 残差ベクトルの計算時に大きな誤差を生じることによると思われる。実際, 残差ベクトルの計算時においてロトキン行列の行列要素を倍精度長まで正しく与えて逐次法による計算を行えば, 表 3-15, 16 と同じく収束解での誤差は零になる。但し, 単精度の $N=7, 8$ では $ILL=1$ が出力され, 逐次法は収束しないと判定された。 $N=8$ では 1 回の計算当りの有効数字—表 3-4 の $-\log f$ に相当する値で -0.0939 —が 1 桁もないので当然であろうが, $N=7$ ではその値が 0.4958 なのでいくらかの精度の向上が期待される場所である。この場合 $d_2/d_1 = 0.508 > 0.5$ のために $ILL=1$ と判定された。そこで, ILL -判定を $d_{k+1} > 2/3 d_k$ として計算してみると, 16 回の繰り返して収束し相対誤差の最大値 3.406₈ が得られた。このときの 1 回当りの精度の向上は大略 0.4960 程である。ただ, 第 1 回目の値が 0.2942 と小さくて $ILL=1$ となっている。このようではあるが, ロトキン行列は次数の小さい割には条件数が非常に大きく, 又 $N=7$ では殆んど有効数字がないということを考慮すれば, ILL -判定が厳し過ぎるものではないと結論できる。

複素行列に関するテストは今回は省略したが, 基本的には実行列の場合と同様と考えられるので, サブルーチンの使用に当たっても同じ配慮が必要である。

表 3-14

		単位 (sec)											
		N		50		100		200		300		400	
$O_N^{(1,P)}$	RLGDTS		0.08		0.62		4.71		16.40		43.57		
	RLGDTD		0.22	0	1.69	0	13.06	0	43.84	0	121.9	0	
	RLGSLS		0.22		1.71		13.24		51.55		145.0		
	RLGSLD		0.64		4.81		37.38		125.6		365.8		
$\bar{O}_N^{(1,P)}$	RLGDTS		0.09	25	0.64	50	4.79	100	16.55	150	44.10	200	
	RLGDTD		0.22		1.72		13.39		44.22		122.4		
	RLGSLS		0.22		1.72		13.30		51.80		145.4		
	RLGSLD		0.65		4.88		37.73		126.5		351.3		
T_N^2	RLGDTS		0.09	47	0.66	98	4.89	196	—	298	—	396	
	RLGDTD		0.23		1.84		14.07		46.75		129.4		
	RLGSLS		0.23		1.77		13.57		—		—		
	RLGSLD		0.66		4.92		37.69		125.7		369.2		

表 3 - 15

EPS = 2.980 -8

	N \ K	3			4			5	
		1	2	3	1	2	3	1	2
絶対誤差	m	7.153 -7	0.0	0.0	1.454 -5	0.0	0.0	1.433 -3	0.0
	f	3.284 -7	0.0	0.0	6.179 -6	0.0	0.0	5.212 -4	0.0
相対誤差	m	3.166 -7	0.0	0.0	1.788 -6	0.0	0.0	3.632 -5	0.0
	f	2.368 -7	0.0	0.0	1.250 -6	0.0	0.0	2.769 -5	0.0
d_K		1.0	2.384 -7	6.209 -9	1.0	1.673 -6	9.934 -9	1.0	3.137 -5
$1/2 d_{K-1}$			0.5	1.192 -7		0.5	8.365 -7		0.5

3	6				7				
	1	2	3	4	1	2	3	4	5
0.0	6.876 -2	3.433 -5	0.0	0.0	2.745	2.370 -2	2.060 -4	3.815 -6	0.0
0.0	2.227 -2	1.071 -5	0.0	0.0	8.348 -1	7.208 -3	6.219 -5	9.447 -7	0.0
0.0	5.990 -4	2.857 -7	0.0	0.0	1.256 -2	1.092 -4	9.598 -7	2.838 -8	0.0
0.0	4.626 -4	2.216 -7	0.0	0.0	8.544 -3	7.380 -5	6.382 -7	1.045 -8	0.0
7.451 -9	1.0	4.805 -4	2.355 -7	1.210 -8	1.0	9.071 -3	7.830 -5	6.755 -7	1.161 -8
1.569 -5		0.5	2.402 -4	1.177 -7		0.5	4.535 -3	3.915 -5	3.377 -7

8								
1	2	3	4	5	6	7	8	9
8.992 +2	8.389 +1	7.828	7.302 -1	6.812 -2	6.348 -3	4.883 -4	1.221 -4	1.526 -5
2.482 +2	2.316 +1	2.161	2.016 -1	1.882 -2	1.749 -3	1.520 -4	2.360 -5	2.791 -6
1.264 -1	1.190 -2	1.110 -3	1.036 -4	9.668 -6	9.423 -7	8.073 -8	2.838 -8	2.755 -8
9.292 -2	8.673 -3	8.092 -4	7.550 -5	7.049 -6	6.607 -7	5.686 -8	9.396 -9	7.077 -9
1.0	1.204 -1	1.111 -2	1.038 -3	9.684 -5	9.035 -6	8.402 -7	8.811 -8	1.808 -8
	0.5	6.021 -2	5.557 -3	5.191 -4	4.842 -5	4.517 -6	4.201 -7	4.405 -8

表 3 - 16

EPS = 8.674 -19

	N \ K	5			6			7	
		1	2	3	1	2	3	1	2
絶対誤差	m	1.707 -14	0.0	0.0	3.996 -13	0.0	0.0	1.102 -10	0.0
	f	6.303 -15	0.0	0.0	1.306 -13	0.0	0.0	3.293 -11	0.0
相対誤差	m	4.450 -16	0.0	0.0	3.256 -15	0.0	0.0	7.647 -13	0.0
	f	3.542 -16	0.0	0.0	2.519 -15	0.0	0.0	3.589 -13	0.0
d_K		1.0	4.324 -16	2.168 -19	1.0	2.852 -15	3.525 -19	1.0	4.290 -13
$1/2 d_{K-1}$			0.5	2.162 -16		0.5	1.426 -15		0.5

	8			9			10		
3	1	2	3	1	2	3	1	2	3
0.0	1.678 ⁻⁷	0.0	0.0	2.677 ⁻⁶	0.0	0.0	1.017 ⁻⁵	7.105 ⁻¹⁵	0.0
0.0	4.665 ⁻⁸	0.0	0.0	7.060 ⁻⁷	0.0	0.0	2.490 ⁻⁶	1.881 ⁻¹⁵	0.0
0.0	3.638 ⁻¹¹	0.0	0.0	5.802 ⁻¹⁰	0.0	0.0	1.386 ⁻⁹	8.665 ⁻¹⁹	0.0
0.0	1.790 ⁻¹¹	0.0	0.0	3.198 ⁻¹⁰	0.0	0.0	7.631 ⁻¹⁰	6.055 ⁻¹⁹	0.0
3.460 ⁻¹⁹	1.0	1.827 ⁻¹¹	2.958 ⁻¹⁹	1.0	3.173 ⁻¹⁰	3.776 ⁻¹⁹	1.0	7.847 ⁻¹⁰	8.281 ⁻¹⁹
2.145 ⁻¹³		0.5	9.137 ⁻¹²		0.5	1.586 ⁻¹⁰		0.5	3.924 ⁻¹⁰

表 3-17

			3	4	5	6	7	8	9	10
絶対 誤差	単	<i>m</i>	5.341 ⁻⁵	5.579 ⁻²	7.282 ⁺¹	1.168 ⁺⁴	4.131 ⁺⁷	4.285 ⁺⁹		
		<i>f</i>	2.562 ⁻⁵	2.525 ⁻²	2.634 ⁺¹	3.777 ⁺³	1.245 ⁺⁷	1.186 ⁺⁹		
	倍	<i>m</i>	1.110 ⁻¹⁵	1.904 ⁻¹²	1.118 ⁻⁹	1.478 ⁻⁷	6.080 ⁻⁴	7.491 ⁻¹	4.184 ⁺²	1.870 ⁺⁵
		<i>f</i>	5.304 ⁻¹⁶	8.586 ⁻¹³	4.061 ⁻¹⁰	4.783 ⁻⁸	1.832 ⁻⁴	2.072 ⁻¹	1.107 ⁺²	4.572 ⁻⁴
相対 誤差	単	<i>m</i>	3.576 ⁻⁷	2.201 ⁻⁵	8.818 ⁻⁴	4.074 ⁻³	6.336 ⁻¹	1.848		
		<i>f</i>	3.011 ⁻⁷	1.374 ⁻⁵	5.598 ⁻⁴	3.230 ⁻³	3.595 ⁻¹	1.231		
	倍	<i>m</i>	6.939 ⁻¹⁸	7.173 ⁻¹⁶	1.309 ⁻¹⁴	5.255 ⁻¹⁴	9.465 ⁻¹²	3.735 ⁻¹⁰	6.954 ⁻⁹	1.004 ⁻⁷
		<i>f</i>	5.913 ⁻¹⁸	4.616 ⁻¹⁶	8.652 ⁻¹⁵	4.147 ⁻¹⁴	5.301 ⁻¹²	2.143 ⁻¹⁰	4.029 ⁻⁹	5.732 ⁻⁸

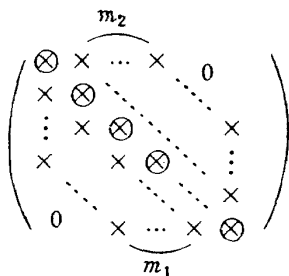
4章 帯状行列のLU分解と方程式系の解法

4.1 方法の基礎

ここでいう帯状行列とは、行列 A を $A=(a_{ij})$ としたとき

$$a_{ij} = 0 \quad i > j + m_1 \quad \text{又は} \quad j > i + m_2 \quad (4.1)$$

となる行列のことである。この行列は対角線より下に m_1 個、上に m_2 個の非零要素をもっている。これを図示すると次図のようになる。(⊗は対角要素, ×は非零非対角要素を表わす。以下同じ)



本章では前章で述べたうちの partial pivoting を行なうドット積法を採用するが、行列が帯状であるという特殊性を考慮してアルゴリズムを若干修正する。又 A が対称行列であるとき、「 $A-\lambda I$ の n 個の主小行列式 $p_i \quad i=1, \dots, n$ と $p_0=1$ において、引き続き p_i の符号が一致する回数は、 λ より大きい行列 A の固有値の数に等しい」という性質があるので、この回数がわかるように pivoting の実行の仕方を修正したアルゴリズムによる方法も採用した。

行列の固有ベクトルは inverse iteration によって求める。即ち、近似固有値 λ が与えられたとき、この固有値に対応する固有ベクトルは、系

$$(A-\lambda I)z_{r+1} = x_r \quad a_{r+1} = \|z_{r+1}\|_{\infty} = \max_i |(z_{r+1})_i|$$

$$x_{r+1} = z_{r+1} / a_{r+1} \quad (4.2)$$

を解くことによって得られる。各 x_r には必ず自明な 1 という要素があるわけだが、その位置に対応する z_{r+1} の要素を β_{r+1} とすると、

$$\lambda + 1/\beta_{r+1} \quad (4.3)$$

は改良された固有値を与えてくれる。この iteration の終了のための便利な基準は、

$$\|Ax_r - (\lambda + 1/\beta_r)x_r\|_\infty \leq \delta \quad (4.4)$$

である。この式は、固有値と固有ベクトルに対する現近似値に関する剰余ベクトルが、 $\|A\|_\infty = \max_i \sum_j |a_{ij}|$ と計算機精度に依存して予め与えられた許容誤差 δ より小さいことを表わしている。このままの形では毎回 Ax_r という形のベクトルを計算することになり、計算量が多くなるので、これを

$$\|(x_{r-1} - z_r/\beta_r)/a_r\|_\infty \leq \delta \quad (4.5)$$

という形に変形して使用する。但し、ここで留意すべきことは、 z_r というのが $(A - \lambda I)z_r = x_{r-1}$ の計算解であり、 λ が A の近似固有値なので $A - \lambda I$ の条件数は通常非常に大きく、従って計算解 z_r の精度は良くない。それ故、(4.5)式を使用するには若干その正当性を保証する必要がある。(4-3 参照)又、改良された λ を用いて二度目の分解を行なうのが不経済である程度に、 λ が十分良い近似であると仮定する。

始めに述べた二つの分解のどちらでも $(A^T - \lambda I)y = b$ という系を解くことができるので、固有値 λ に対応する左固有ベクトルを inverse iteration によって求めることができる。左右の固有ベクトルが異なるのは A が非対称のときであり、後者の分解は A が対称なときに有用だから、左固有ベクトルを求める目的のためには前者の分解を用いる。

A が対称なとき、固有ベクトル x を用いて

$$x^T Ax / x^T x = \lambda + x^T (Ax - \lambda x) / x^T x \quad (4.6)$$

で定義されるレーレー商によって固有値を改良することができる。このとき、 $Ax - \lambda x$ の計算は倍精度で行なう。実際の計算では、(4.3)式を用いるよりこちらの方が優れている²⁾ A が非対称のときには、右固有ベクトル x と左固有ベクトル y とによって

$$y^T Ax / y^T x = \lambda + y^T (Ax - \lambda x) / y^T x$$

と定義される一般化されたレーレー商を用いて固有値を改良する。

4.2 計算方法とサブルーチンの説明

4.2.1 帯状行列に対する通常のドットル分解の手続き

これには $n-1$ 回の主ステップがあって、第 r ステッ

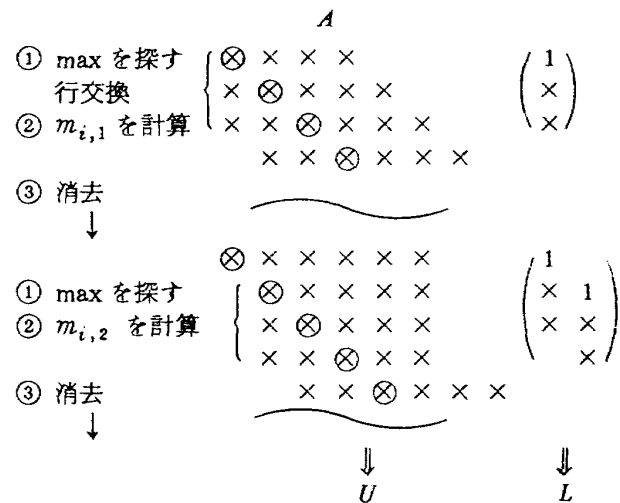
プで行われることは A の第 r 列の対角線より下にある要素を消去することである。消去されるべき非対角要素の数は、最後の m_1 回のステップを除いては m_1 個である。最後の m_1 回は一つずつ少なくなっていく。これを整理した形で書くと、 $l = \min(n, r + m_1)$ として

(1) $|a_{i_r, r}| = \max_{r \leq i \leq l} |a_{i, r}|$ となる最小の i_r を探して、 i_r 行と r 行とを入れ換える。

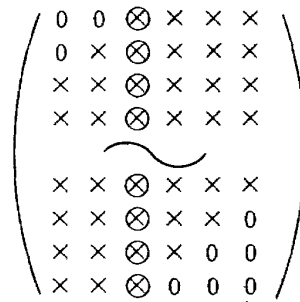
(2) $m_{i, r} = a_{i, r} / a_{i_r, r}$ を計算する。 $r+1 \leq i \leq l$, $|m_{i, r}| \leq 1$

(3) $a_{i, j} = a_{i, j} - m_{i, r} \times a_{r, j}$ を計算する。 $r+1 \leq i \leq l$

こうして $n-1$ 回のステップの後には、 A に生成される行列が U であり、 $m_{i, r}$ によって作られる行列を L , i_r を操作して得られる置換行列を P とすると、これで $\tilde{A} = PA$ のドットル分解 $\tilde{A} = LU$ が完成する。これを実際の行列で、 $m_1 = 2, m_2 = 3$ として図示すると次のようになる。



このように普通の行列表示を用いると、自明な零要素のために無駄なメモリーを使うことになる。そこでこれを次のようにして圧縮する。($m_1 = 2, m_2 = 3$)

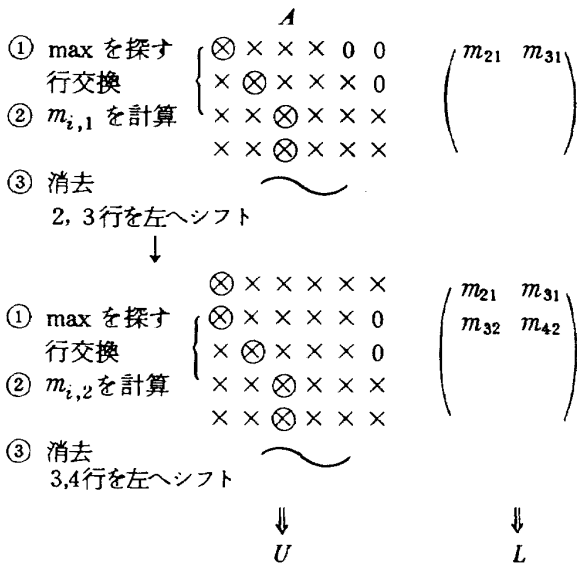


対角線より 対角線より
下の部分 上の部分

上図のような形で A をインプットし、これをルーチンの始めで

$$\begin{pmatrix} \otimes & \times & \times & \times & 0 & 0 \\ \times & \otimes & \times & \times & \times & 0 \\ \times & \times & \otimes & \times & \times & \times \\ \times & \times & \otimes & \times & 0 & 0 \\ \times & \times & \otimes & 0 & 0 & 0 \end{pmatrix}$$

という形につめ直す。そして、第 r 列を消去するとき、第 $r+1$ 行から第 l 行までを一つずつ左に送っていく。そうすると消去される列は常に第 1 列の第 r 行から第 l 行にあることになる。これを図示すると次のようになる。



L についても自明な 1 はストアせず、非対角要素だけを前図右のようにストアしていく。これからもわかるように、 L は対角線より下に m_1 個の非零要素をもつ下三角行列、 U は行交換を行なうので一般には対角線より上に $m_1 + m_2$ 個の非零要素をもつ上三角行列である。この方法で LU 分解するのが BNDTIS、それによって系を解くのが BNSLIS である。

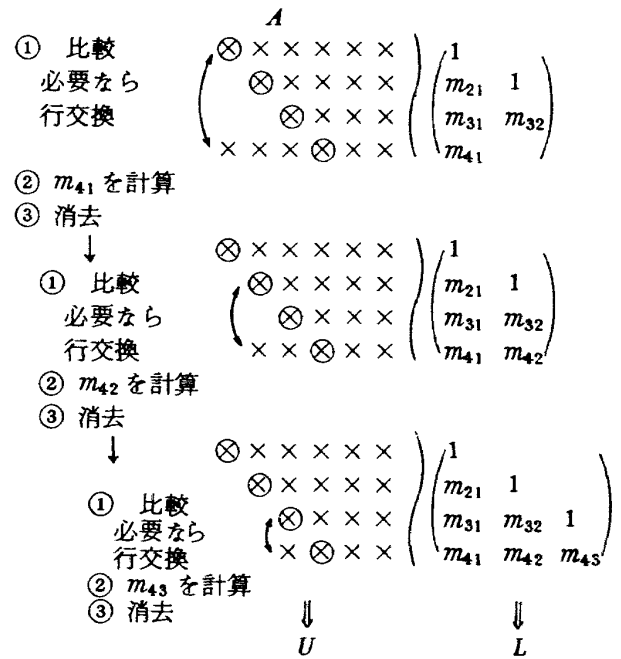
4.2.2 主小行列式の値 p_r がわかる pivoting 方法の手続き

簡単のためにこの方法を稠密な行列について説明する。この方法も $n-1$ 回の主ステップより成っている。第 r ステップの始めには、元の行列の第 1 行から第 r 行までが次図のように変形されており ($n=5, r=3$),

$$\left. \begin{matrix} \otimes & \times & \times & \times \\ \otimes & \times & \times & \times \\ & \otimes & \times & \times \\ \times & \times & \otimes & \times \\ \times & \times & \times & \otimes \end{matrix} \right\} \text{変化せず}$$

第 r ステップでは、この変形された r 個の行に元の行列の第 $r+1$ 行をつけ加え、 r 回の従ステップによって上図のような形 ($r=4$ にする) に変形する。この従ループの第 i ステップは次のように行われる。 $i=1, \dots, r$

- (1) $|a_{r+1,i}| > |a_{i,i}|$ ならば $r+1$ 行と i 行を入れ換える。
- (2) $m_{r+1,i} = a_{r+1,i}/a_{i,i}$ を計算する。 $|m_{r+1,i}| \leq 1$
- (3) $a_{r+1,j} = a_{r+1,j} - m_{r+1,i} \times a_{i,j}$ を計算する。これを $r=3$ として図示すると次のようになる。



そうすると、 $r+1$ 番目の主小行列式の値 p_{r+1} は

$$p_{r+1} = (-1)^{\sigma_{r+1}} \prod_{i=1}^{r+1} a_{i,i}$$

で与えられる。ここで σ_{r+1} は、始めの r 回の主ステップにおける行交換の総数である。

行列が帯状のときには、4.2.1 と同じように圧縮した形にして、このアルゴリズムに同様の変更を加える。このとき始めの方の数スラップを除いて、主ステップで対象となる行の数は $m_1 + 1$ 個であって、始めの方の $r - m_1$ 個の行は消去に関係しない。従って、 p_{r+1} の因子のうち始めの $r - m_1$ 個は、 p_r のそれと同じものである。又、固有値の数を求めるには、 p_r の符号変化がわかれば良いだけだから、 p_r の値そのものを求めるより p_{r+1}/p_r の符号を決める方が経済的である。なお、この場合の行交換の情報の保存には、第 r 主ステップで、 r 行と i 行を交換したかどうかという形で行なう必要がある。そのためには、 $n \times m_1$ の大きさの 2 次元論理型アレイが必要になる。従って、一つの論理変数のために 1 語を必要とする計算機 (現在の FACOM 230-75 はそうである) では、領域的には不経済である。一方、一つの論理変数を 1 ビットで表わす計算機ではそのような不利さが減る。(導入された FACOM APU はそうになっていて、それが必要と

する語数は $\lceil \frac{n \times m_1}{32} \rceil + 1$ 語である) なお、前者の方法では帯状であるとなしに拘わらず n 語がこのために必要である。この方法で LU 分解するのが BNDTBS, それによって系を解くのが BNSLBS である。

4.2.3 その他

上記1, 2のどちらの場合にも, inverse iteration を行ったり, 行列式の値を計算するだけであつたりするときには, ピボット要素として零がでてきてもルーチンを終わらせる必要がない。さらにいえば, inverse iteration は λ が真の固有値に近い程効果が有り, このときにはピボット要素が零に近くなるのは普通なことである。一方, 方程式を解くときには, ピボット要素が零になるということは, $A - \lambda I$ が非正則であることを意味し, このときにはルーチンを終わらせる必要がある。このために, 分解を行うサブルーチンに引き数を一つ加えて, ピボット要素が零になったときルーチンを終わせないかの選択を行わせる。なお, ルーチンを終わせないときには, 零になったピボット要素は $2^{-t+1} \|A\|_\infty$ という値で置きかえる。(t は仮数部桁数, $\|A\|_\infty = \max_i \sum_j |a_{ij}|$)

inverse iteration を行なうとき, その初期ベクトルの選択はいささか難しい問題である。右固有ベクトルを計算するときには, 全要素が1のベクトル d に対して, 逆進過程

$$Ux = d$$

を計算することをその出発点とする。即ち, 前進過程を省略するわけである。このようにすれば初期ベクトルを Ld として inverse iteration を1回行ったことに相応し, 又異なる固有値には異なる LU 分解が対応するので異なる初期ベクトルで出発することになり, この点での不都合はない。([2] を参照)。このために, $LUx = b$ を解くルーチンに引き数を一つ加えて, 前進過程を省略するかしないかの選択を行わせる。左固有ベクトルの計算には, 計算されてきた右固有ベクトルを初期近似としてとる。

レ-レ-商を用いて固有値の値を改良するとき, $Ax - \lambda x$ を正確に計算する必要があるので, この部分は倍精度で行なう。

4.2.1, 2で述べたどちらの方法も, 任意の帯状行列に対して適用できるが, 普通には $m_1 + m_2 \ll n$ のときのみ使用する。 A が正値対称帯状行列であつて方程式系を解くだけのときには, 「1次方程式系の解法I」にあ

るサブルーチン BCLDTS, BCLSLTS を使用する。 inverse iteration で固有ベクトルを求めようとするなら, 前記サブルーチンは使用できない。というのは, $A - \lambda I$ が一般的には正定値とならないからである。なお, どちらの分解でも, 最初に A の対角要素から λ を引くので, A そのものを分解したいときには $\lambda = 0$ とする。

4.3 数値的議論

inverse iteration における収束判定式 (4.5) について少し述べる。 λ が固有値に十分近いとき — inverse iteration を行うとき, このことは普通である —, $(A - \lambda I) z_r = x_{r-1}$ の計算解 z_r は真の値に比べて相当違っている。(4.4, テスト結果参照)しかし, ウィルキンソンの解析³⁾によると, z_r は式

$$(A + E_r - \lambda I) z_r = x_{r-1}$$

を満足し, $\|E_r\|$ の値は十分小さい。そうすると,

$$\|(x_{r-1} - \beta_r^{-1} z_r) / \alpha_r\|_\infty \leq \delta$$

より

$$\|(A + E_r) z_r / \alpha_r - (\lambda + \beta_r^{-1}) z_r / \alpha_r\|_\infty \leq \delta$$

が得られ, 結局

$$\|(A + E_r) x_r - (\lambda + \beta_r^{-1}) x_r\|_\infty \leq \delta$$

となる。 x_r は, その絶対値最大の要素が1となるように正規化されている。この式は, $\lambda + \beta_r^{-1}$ と x_r が行列 $A + E_r$ の固有値と固有ベクトルであることを示している。ここで,

$$\|F_r\| \leq \|E_r\| + \delta$$

レ-レ-商による固有値 λ_p の計算誤差は,

$$a + 2^{-2t} (m_1 + m_2 + 1) \max |a_{ij}| / b \cos \theta_p$$

を越えない。ここで, t は仮数部桁数で,

$$a = \max_i |\lambda_p - \lambda_i| \\ |\lambda_p - \lambda_i| < 2^{-t}$$

$$b = \min_i |\lambda_p - \lambda_i| \\ |\lambda_p - \lambda_i| \geq 2^{-t}$$

であり, θ_p は右固有ベクトルと左固有ベクトルのなす角である。この θ_p は行列の固有値問題の安定性に関わりをもつものである。

なお, 方程式系の誤差評価についても3.3と同様の評価が得られる。^{1), 2), 3)}

サブルーチン I

(1) BNDTIS, BNDTID

このサブルーチンは、帯状行列 A に対して $A-\lambda I$ を 4.2.1 で述べた方法で LU 分解 (ドットル分解) し、副産物として $A-\lambda I$ の行列式の値を与える。

元の A の値は失われる。

呼び出し形式

BNDTIS ($N, M1, M2, R, L, A, KA, F, INT, D, MD, ILL$)

インプット

N : 行列 A の次数, $N > 2$

整数又は整数型変数名

$M1, M2$: 4.1 で定義した行列 A の帯巾

$1 \leq M1, M2 \leq N-1$

整数又は整数型変数名

R : 前記 λ の値を与える。

A そのものを分解したいときには 0.0 を与える。

実数, 又は実数型変数名

L : ピボット要素が零になったときのルーチンの処理の仕方を定める (4.2.3 を参照)

整数又は整数型変数名

$= 1$ ピボットが零になってもルーチンを終らない

$\neq 1$ ピボットが零になればルーチンを終る。

A : $N \times N$ 行列 A の要素を $N \times (M1+M2+1)$ の 2 次元アレイとしてストアする。

配列の大きさが KA の実数型アレイ。

配列の仕方については図 4-1 を参照。

KA : アレイ A, F の大きさを指定, $KA \geq N$

整数又は整数型変数名

アウトプット

A : ドットル分解した結果の上三角行列 U の要素が図 4-1 のようにストアされている。

F : ドットル分解した結果の下三角行列 L の非対角要素が $N \times M1$ の 2 次元アレイとして、図 4-1 のようにストアされている。

配列の大きさが KA の実数型アレイ。

INT : 行交換の情報を保存している。

大きさ N の 1 次元整数型アレイ。

D, MD : $A-\lambda I$ の行列式の値を $D \times 2^{MD}$ の形で与える。

D は実数型変数名で, $1/2 \leq |D| \leq 1$

ILL : 整数型変数名

$= 0$ 正常に終了したとき。

$= K$ $L \neq 1$ で第 K 番目のピボット要素が零になったとき, $1 \leq K \leq N$

$= 30000$ 入力パラメーターに誤りがあったとき。

エラー処理

入力パラメーターに誤りがあった場合には, ILL 値をセットしただけでリターンする。従って, このときにはアレイ A の内容は元のままである。

$L \neq 1$ の指定があつて第 K 番目のピボット要素が零になった場合には, ILL 値をセットし, それより先の計算は打ち切ってリターンする。従って, このときにはアレイ A の内容は元の値ではない。

詳細説明

インプット A 及びアウトプット A, F の配列の仕方を $N=5, M1=2, M2=1$ として例示する。

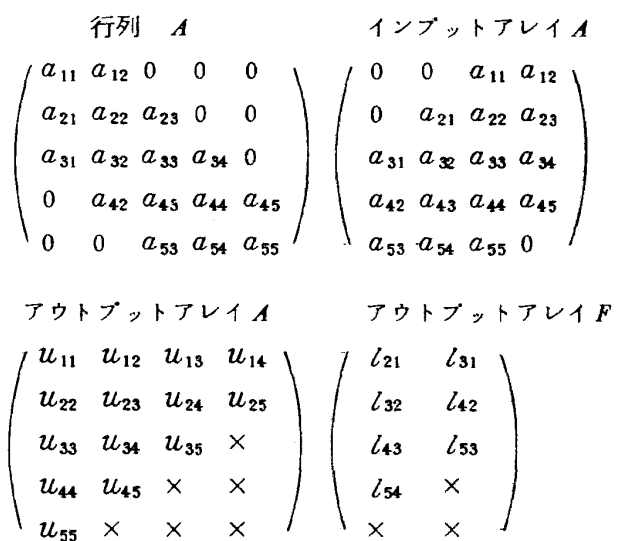


図 4-1

行交換の情報は次のような形で保存されている。行列 A に対し, 第 K 行と第 $INT(K)$ 行との交換を $K=1, \dots, N$ について行って得られる行列を \tilde{A} とする。(即ち, 互換の積 $(N, INT(N)) \dots (1, INT(1))$ を σ としたとき, 行列 A の行に置換 σ を施すと得られる) この \tilde{A} をドットル分解した場合の L, U がここに得られるものである。但し, L についてもその対角要素を除く下三角部分に同様の行交換を行ったものを \tilde{L} としたとき, $\tilde{A} = \tilde{L}U$ となる。(4.4, テスト結果参照)

通常の方程式 $AX=B$ を解くために A を LU 分解するときには $R=0.0, L=0$ とすればよい。

BNDTID を用いるときには, R, A, F, D を倍精度実数型として指定する。

(2) BNSLIS, BNSLID

このサブルーチンは、帯状行列 A を係数行列とする方程式 $AX=B$ を解くもので、 A を $A=LU$ とドットリトル分解し (BNDTIS), ついで $LY=B$, $UX=Y$ を解いて X を求める (BNSLIS)。

前項のサブルーチン(1)からのアウトプット A, F, INT , が本サブルーチンのインプット A, F, INT になるので、このサブルーチンを使用する前に必ず(1)のサブルーチンを使用しておく。もしくはそれと同等な手続に従って A, F, INT に図4-1のような情報をセットする。一度そのようにセットされれば、このサブルーチンを何度使用してもよい。

解 X は B がストアされていた場所にストアされるので、 B の元の値は失われる。

呼び出し形式

BNSLIS ($N, M1, M2, NR, L, A, KA, F, INT,$
 B)

インプット

N : 行列 A の次数, $N \geq 2$

整数又は整数型変数名

$M1, M2$: 4.1 で定義した行列の帯巾

$1 \leq M1, M2 \leq N-1$

整数又は整数型変数名

NR : 方程式の右辺の数, $NR \geq 1$

整数又は整数型変数名

L : 前進過程を省略するかどうかを決める。

= 0 前進過程を省略しない。

通常に方程式を解く。

≠ 0 前進過程を省略する。

A : 係数行列 A をドットリトル分解して得られる上三角行列 U の要素を図4-1のようにストアする。

BNDTIS からのアウトプット A でよい。

大きさ KA の, $N \times (M1 + M2 + 1) - 2$ 次元実数型アレイ。

KA : アレイ A, F, B の大きさを指定, $KA \geq N$
整数又は整数型変数名

F : 係数行列 A をドットリトル分解して得られる下三角行列 L の要素を図4-1のようにストアする。

BNDTIS からのアウトプット F でよい。

大きさ KA の, $N \times M1 - 2$ 次元実数型アレイ。

INT : 係数行列 A をドットリトル分解する際の行交換の情報ストアする。

BNDTIS からのアウトプット INT でよい。

大きさ N の 1 次元整数型アレイ。

B : 方程式の右辺である $N \times NR$ 行列 B の要素を通常行列表示と同じようにストアする。

大きさ KA の 2 次元実数型アレイ。

アウトプット

B : 方程式 $AX=B$ の解 X がストアされている。

エラー処理

インプットパラメーターに誤りがあったときには何もせずにそのままリターンする。ILL 値のセットも行わない。

詳細説明

通常の方程式 $AX=B$ を解くには $L=0$ とする。

BNSLID を用いるときには、 A, F, B を倍精度実数型として指定する。

サブルーチン II

(1) BNDTBS, BNDTBD

このサブルーチンは、帯状行列 A に対して $A - \lambda I$ を 4.2.2 で述べた方法で LU 分解 (ドゥリトル分解) し、副産物として $A - \lambda I$ の行列式の値を与える。さらに、 A が対称行列のとき、後で述べるパラメーター L を 1 にすれば、 λ より大きい A の固有値の数がわかる。(λ そのものが A の固有値に近くなければ、 L は 1 でなくともよい) 元の A の値は失われる。

呼び出し形式

BNDTBS($N, M1, M2, R, L, A, KA, F, INT, NE, D, MD, ILL$)

インプット

N : 行列 A の次数, $N \geq 2$

整数型変数名

$M1, M2$: 4.1 で定義した行列 A の帯巾

$1 \leq M1, M2 \leq N-1$

R : 前記 λ の値を与える。

A そのものを分解したいときには、0.0 を与える。

実定数又は実数型変数名

L : ピボット要素が零になったときのルーチンの処理の仕方を定める。(4.2.3 を参照)

整数型変数名

= 1 ピボットが零になってもルーチンを終わらない。

≠ 1 ピボットが零になればルーチンを終わる。

A : $N \times N$ 行列 A の要素を $N \times (M1 + M2 + 1)$ の 2 次元アレイとしてストアする。

配列の大きさが KA の実数型アレイ。

配列の仕方については図 4-2 を参照

KA : アレイ A, F の大きさを指定, $KA \geq N$

整数型変数名

アウトプット

A : ドゥリトル分解した結果の上三角行列 U の要素が図 4-2 のようにストアされている。

F : ドゥリトル分解した結果の下三角行列 L の非対角要素が $N \times M1$ の 2 次元アレイとして、図 4-2 のようにストアされている。

配列の大きさが KA の実数型アレイ。

INT : 行交換の情報を保存している。

大きさが KA の $N \times M1$ 2 次元論理型アレイ。

NE : λ より大きい A の固有値の数を与える。

整数型変数名

D, MD : $A - \lambda I$ の行列式の値を $D \times 2^{MD}$ の形で与える。

D は実数型変数名で、 $1/2 \leq |D| \leq 1$

ILL : 整数型変数名

= 0 正常に終了したとき。

= K $L \neq 1$ で第 K 番目のピボット要素が零になったとき、 $1 \leq K \leq N$

= 30000 入力パラメーターに誤りがあったとき。

エラー処理

入力パラメーターに誤りがあった場合には、 ILL 値をセットしただけでリターンする。従って、このときにはアレイ A の内容は元のままである。

$L \neq 1$ の指定があって第 K 番目のピボット要素が零になった場合には、 ILL 値をセットし、それより先の計算は打ち切ってリターンする。従って、このときにはアレイ A の内容は元の値ではない。

詳細説明

インプット A 及びアウトプット A, F の配列の仕方を $N=5, M1=2, M2=1$ として例示する。

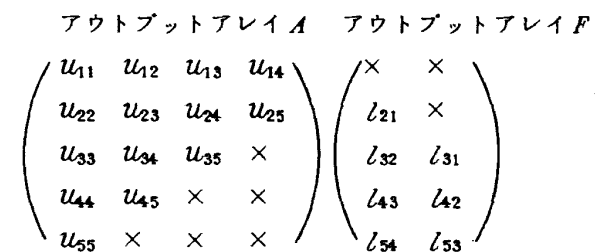
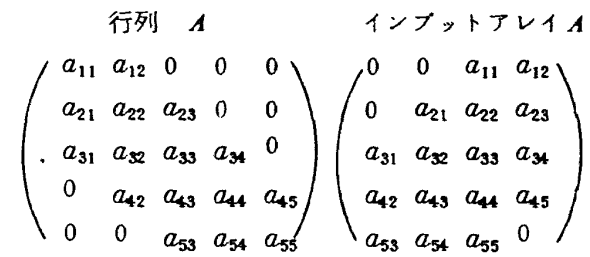


図 4-2

行交換の情報は次のような形で保存されている。行列 A に対し、 $INT(I, J) = \text{TRUE}$ のとき第 I 行と第 $I-J$ 行との交換を、 $J = \min(I-1, M1), 1, -1$; $I=2, N$ について行って得られる行列を \tilde{A} とする。この \tilde{A} をドゥリトル分解した場合の L, U がここに得られるものである。但し、 L についてもその対角要素を除く下三角部分に同様の行交換を行ったものを \tilde{L} としたとき、 $\tilde{A} = \tilde{L}U$ となる。(4.4, テスト結果参照)

通常の方程式 $AX=B$ を解くために A を LU 分解する

ときには $R=0.0$, $L=0$ とすればよい。

BNDTBD を用いるときには, R, A, F, D を倍精度実数型として指定する。

(2) BNSLBS, BNSLBD

このサブルーチンは, 帯状行列 A を係数行列とする方程式 $AX=B$ を解くもので, A を $A=LU$ とドットリトル分解し (BNDTBS), ついで $LY=B$, $UX=Y$ を解いて X を求める (BNSLBS)。

前項のサブルーチン(1)からのアウトプット A, F, INT が本サブルーチンのインプット A, F, INT になるので, このサブルーチンを使用する前に必ず(1)のサブルーチンを使用しておく。もしくはそれと同等な手続に従って A, F, INT に図4-2のような情報をセットする。一度そのようにセットされれば, このサブルーチンを何度使用してもよい。

解 X は B がストアされていた場所にストアされるので, B の元の値は失われる。

呼び出し形式

BNSLBS ($N, M1, M2, NR, L, A, KA, F, INT,$
 B)

インプット

N : 行列 A の次数, $N \geq 2$

整定数又は整数型変数名

$M1, M2$: 4.1 で定義した行列の帯巾

$1 \leq M1, M2 \leq N-1$

整定数又は整数型変数名

NR : 方程式の右辺の数, $NR \geq 1$

整定数又は整数型変数名

L : 前進過程を省略するかどうかを決める。

= 0 前進過程を省略しない。

通常に方程式を解く。

$\neq 0$ 前進過程を省略する。

A : 係数行列 A をドットリトル分解して得られる上三角行列 U の要素を図4-2のようにストアする。

BNDTBS からのアウトプット A でよい。

大きさ KA の, $N \times (M1+M2+1) - 2$ 次元実数型アレイ。

KA : アレイ A, F, INT, B の大きさを指定,

$KA \geq N$

整定数又は整数型変数名

F : 係数行列 A をドットリトル分解して得られる下三角行列 L の要素を図4-2のようにストアする。

BNDTBS からのアウトプット F でよい。

大きさ KA の, $N \times M1 - 2$ 次元実数型アレイ。

INT : 係数行列 A をドットリトル分解する際の行交換の情報をストックする。

BNDTBS からのアウトプット INT でよい。大き

さ KA の, $N \times M1 - 2$ 次元論理型アレイ。

B : 方程式の右辺である $N \times NR$ 行列 B の要素を通常の行列表示と同じようにストアする。

大きさ KA の 2次元実数型アレイ。

アウトプット

B : 方程式 $AX=B$ の解 X がストアされている。

エラー処理

インプットパラメーターに誤りがあったときには何もせずにそのままリターンする。ILL値のセットも行わない。

詳細情報

通常方程式 $AX=B$ を解くには $L=0$ とする。

BNSLBD を用いるときには, A, F, B を倍精度実数型として指定する。

サブルーチン III

(1) BUIVTS, BUIVTD

このサブルーチンは、主として非対称帯状行列 A の複数の右及び左固有ベクトルを (4.2)-式の *inverse iteration* 法によって計算するものである。入力データとして近似固有値を必要とする。この値は、一般化されたレーレー商を計算することによって改良された近似値に変更される。元の A の値は保存される。

呼び出し形式

BUIVTS ($N, M1, M2, NR, A, KA, NU, R, IR,$
 $IL, VR, VL, W1, W2, IW, W3, W4,$
 ILL)

インプット

N : 行列 A の次数, $N \geq 2$

整定数又は整数型変数名

$M1, M2$: 4.1 で定義した行列 A の帯巾

$1 \leq M1, M2 \leq N-1, M1+M2-1 < N$

整定数又は整数型変数名

NR : 計算すべき固有ベクトルの数, $NR \geq 1$

整定数又は整数型変数名

A : $N \times N$ 行列 A の要素を $N \times (M1+M2+1)$ の 2次元アレイとしてストアする。

配列の大きさが KA の実数型アレイ。

配列の仕方については図 4-3 を参照。

KA : アレイ $A, VR, VL, W1, W2, V$ の大きさを指定, $KA \geq N$

整定数又は整数型変数名

NU : イタレーション回数の上限を指定, $NU \geq 1$

整定数又は整数型変数名

R : NR 個の近似固有値を与える。

大きさ NR の 1次元実数型アレイ。

アウトプット

R : 改良された近似値がストアされている。

IR, IL : それぞれ右固有ベクトル, 左固有ベクトルの計算に要したイタレーション回数がストアされている。

共に大きさ NR の 1次元整数型アレイ。

VR, VL : それぞれ右固有ベクトル, 左固有ベクトルが列ベクトルの形でストアされている。

共に大きさ KA の, $N \times NR - 2$ 次元実数型アレイ。

$W1, W2, W3$: ワーキングエリア

大きさ KA の 2次元実数型アレイ。

$W1$ は $N \times (M1+M2+1)$, $W2$ は $N \times M1$, $W3$

は $N \times 1$ である。

IW : ワーキングエリア

大きさ N の 1次元整数型アレイ。

$W4$: ワーキングエリア

大きさ N の 1次元実数型アレイ。

ILL : 整数型変数名

= 0 正常終了

= K K 番目以下の右又は左固有ベクトルの計算において, 最初に与えられたイタレーション回数の上限内で収束しなかったものがある。 $1 \leq K \leq NR$

= 30000 入力パラメーターに誤りがあったとき。

エラー処理

入力パラメーターに誤りがあった場合には, ILL 値をセットだけでリターンする。

K 番目の固有ベクトルの計算中に, イタレーション回数が上限値 NU を越えても収束しなかった場合, ILL に K をセットし, $IR(K)$ (もしくは $IL(K)$) には $NU+1$ をセットする。この後, $M > K$ 番目にも同様の事態が生じれば, ILL の値は M になっている。即ち, ILL の値が K であるということは, K 番目はもちろん, それ以下の番号のものの中にも NU 回かかってイタレーションが収束しなかったものがあり得ることを表わす。このとき, 対応する VR (もしくは VL) には最終近似固有ベクトルがストアされており, R にはそれに基づく改良固有値がストアされている。

詳細説明

インプット A の配列の仕方は今までと同じであるが次図に示す。 $N=5, M1=2, M2=1$

$$\begin{array}{cc} \text{行列 } A & \text{インプットアレイ } A \\ \left(\begin{array}{ccccc} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} \\ 0 & 0 & a_{53} & a_{54} & a_{55} \end{array} \right) & \left(\begin{array}{cccc} 0 & 0 & a_{11} & a_{12} \\ 0 & a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} & a_{45} \\ a_{53} & a_{54} & a_{55} & 0 \end{array} \right) \end{array}$$

図 4-3

2次元ワーキングエリア $W3$ の DIMENSION は

$W3(KA, 1)$

であることに注意する。

サブルーチン内で, BNDTIS, BNSLIS を使用している。

BUIVTD を用いるときには, $A, R, VR, VL, W1,$
 $W2, W3, W4$ を倍精度実数型として指定する。

(2) BSIVTS, BSIVTD

このサブルーチンは, 対称帯状行列 A の複数個の固有ベクトルを (4.2)-式の inverse iteration 法によって計算するものである。入力データとして近似固有値を必要とする。この値はレーレー商を計算することによって改良された近似値に変更される。元の A の値は保存される。

呼び出し形式

BSIVTS($N, M, NR, A, KA, NU, R, IB, NE,$
 $VB, W1, W2, LW, W3, W4, ILL$)

インプット

N : 行列 A の次数, $N \geq 2$

整定数又は整数型変数名

M : 4.1 で定義した行列の帯巾 (対称なので,

$M=M1=M2$), $1 \leq M < (N+1)/2$

整定数又は整数型変数名

NR : 計算すべき固有ベクトルの数

整定数又は整数型変数名

A : $N \times N$ 行列 A の要素を $N \times (2M+1)$ の 2 次元アレイとしてストアする。

配列の大きさが KA の実数型アレイ。

配列の仕方については図 4-4 を参照。

KA : アレイ $A, VB, W1, W2, W3, LW$ の大きさを指定, $KA \geq N$

整定数又は整数型変数名

NU : イタレーション回数の上限を指定, $NU \geq 1$

整定数又は整数型変数名

R : NR 個の近似固有値を与える。

大きさ NR の 1 次元実数型アレイ。

アウトプット

R : 改良された近似値がストアされている。

IB : 固有ベクトルの計算に要したイタレーション回数がストアされている。

大きさ NR の 1 次元整数型アレイ。

NE : $NE(I)$ には, $R(I)$ より大きい A の固有値の数がストアされている。

大きさ NR の 1 次元整数型アレイ。

VB : 固有ベクトルが列ベクトルの形でストアされている。

大きさ KA の, $N \times NR - 2$ 次元実数型アレイ。

$W1, W2, W3$: ワーキングエリア

大きさ KA の 2 次元実数型アレイ。

$W1$ は $N \times (2M+1)$, $W2$ は $N \times M$, $W3$ は $N \times 1$ である。

LW : ワーキングエリア

大きさ KA の, $N \times NR - 2$ 次元論理型アレイ。

$W4$: ワーキングエリア

大きさ N の 1 次元実数型アレイ。

ILL : 整数型変数名

= 0 正常終了

= K K 番目以下の固有ベクトルの計算において, 最初に与えられたイタレーション回数の上限内で収束しなかったものがある。

$1 \leq K \leq NR$

= 30000 入力パラメーターに誤りがあったとき。

エラー処理

入力パラメーターに誤りがあった場合には, ILL 値をセットしただけでリターンする。

K 番目の固有ベクトルの計算中に, イタレーション回数が上限値 NU を越えても収束しなかった場合, ILL に K をセットし, $IB(K)$ には $NU+1$ をセットする。この後, $M > K$ 番目にも同様の事態が生じれば, ILL の値は M になっている。即ち, ILL の値が K であるということは, K 番目はもちろん, それ以下の番号のものの中にも NU 回かかってもイタレーションが収束しなかったものがあり得ることを表わす。このとき, 対応する VB には最終近似固有ベクトルがストアされており, R にはそれに基づく改良固有値がストアされている。

詳細説明

インプット A の配列の仕方は今までと同じで, 扱うのが対称行列ではあるが, 全要素をストアする必要がある。 $N=5, M=2$ として図示する。

$$\begin{array}{cc} \text{行列 } A & \text{インプットアレイ } A \\ \left(\begin{array}{ccccc} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} \\ 0 & 0 & a_{53} & a_{54} & a_{55} \end{array} \right) & \left(\begin{array}{ccccc} 0 & 0 & a_{11} & a_{12} & a_{13} \\ 0 & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{42} & a_{43} & a_{44} & a_{45} & 0 \\ a_{53} & a_{54} & a_{55} & 0 & 0 \end{array} \right) \end{array}$$

図 4-4

2 次元ワーキングエリア $W3$ の DIMENSION は

$W3(KA, 1)$

であることに注意する。

サブルーチン内で、BNDTBS, BNSLBS を使用している。

BSIVTD を用いるときには、A, R, VB, W1, W2, W3, W4 を倍精度実数型として指定する。

$$\tilde{A} = \begin{pmatrix} 5 & -4 & 1 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 \\ 0 & 1 & -4 & 6 & -4 \\ 0 & 0 & 1 & -4 & 5 \\ -4 & 6 & -4 & 1 & 0 \end{pmatrix}$$

4.4 数値テスト及び結果

まず始めに、BNDTIS と BNDTBS のアウトプットの内容を実際例で示すことにする。行列 A として B_N^2 を採る。即ち

$$A = \begin{pmatrix} 5 & -4 & 1 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \\ 0 & 1 & -4 & 6 & -4 \\ 0 & 0 & 1 & -4 & 5 \end{pmatrix}$$

で、 $M1=M2=2$ である。

これを BNDTIS でドット分解したときのアウトプット A, F, INT は

$$A = \begin{pmatrix} 5 & -4 & 1 & 0 & 0 \\ -16/5 & 29/5 & -4 & 1 & \times \\ -35/16 & 19/4 & -59/16 & \times & \times \\ -64/35 & 116/35 & \times & \times & \times \\ 9/16 & \times & \times & \times & \times \end{pmatrix}$$

$$F = \begin{pmatrix} -4/5 & 1/5 \\ -7/8 & -5/16 \\ -6/7 & -16/35 \\ -55/64 & \times \\ \times & \times \end{pmatrix} \quad INT = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 5 \\ 5 \end{pmatrix}$$

である。又、BNDTBS でドット分解すれば、アウトプットの A は前記と同じであり、F, INT は

$$F = \begin{pmatrix} \times & \times \\ -4/5 & \times \\ -7/8 & 1/5 \\ -6/7 & -5/16 \\ -55/64 & -16/35 \end{pmatrix} \quad INT = \begin{pmatrix} \times & \times \\ F & \times \\ T & F \\ T & F \\ T & F \end{pmatrix}$$

F : .FALSE.
T : .TRUE.

である。この INT による行交換を A に施せばどちらの場合にも

が得られ、実際にはこの行列がドット分解されている。分解結果の上三角行列はアウトプット A の内容を配置しなおし、下三角行列は F を配置しなおし、ついで対角要素を除く下三角部分に行交換を施せば得られる。その結果

$$\tilde{L} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/5 & 1 & 0 & 0 & 0 \\ 0 & -5/16 & 1 & 0 & 0 \\ 0 & 0 & -16/35 & 1 & 0 \\ -4/5 & -7/8 & -6/7 & -55/64 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 5 & -4 & 1 & 0 & 0 \\ 0 & -16/5 & 29/5 & -4 & 1 \\ 0 & 0 & -35/16 & 19/4 & -59/16 \\ 0 & 0 & 0 & -64/35 & 116/35 \\ 0 & 0 & 0 & 0 & 9/16 \end{pmatrix}$$

が得られ、 $\tilde{A} = \tilde{L}U$ となる。

$B_N, B_N^2, B_N \otimes O_{10}^{(1,d)}, B_N \otimes O_{10}^{(2,d)}$ をテスト行列として逆行列を計算した場合の誤差を表 4-1~8 に与える。4.2.1 (BNDTIS, BNSLIS) と 4.2.2 (BNDTBS, BNSLBS) との方法による差は後 2 者に現われたので、それぞれ別の表にしてある。この差は行交換の順序の違いによって生じるものと思われる。 B_N の場合には全く行交換が行われないので同じ結果になり、 B_N^2 の場合には行交換はどちらも同じ回数だけ行われるが、全く同じ順序に行われるので結果になる。しかし、 $B_N \otimes O_{10}^{(1,d)}$ (行列の次数は $10N$) では前者の方法による行交換が $2N$ であるに対し、後者の方法は $15N \sim 17N$ 、 $B_N \otimes O_{10}^{(2,d)}$ では $4N$ と $18N \sim 24N$ と圧倒的に後者の方法の方が多く、それによる行交換の順序の違いが表 3-5~8 に現われたものと思われる。又、同じテスト行列の前章のサブルーチン RLGDTIS と RLGSLIS とからの相違 (表 3-12, 13 参照) は、ピボット要素の探し方の差に依る。このサブルーチンでは単に対象となっている列の絶対値最大の要素をピボット要素としている。(前章では行のノルムが 1 になるような因子を掛けての最大値をピボット要素とした)

表 4-1

 B_N ; Single

		N	50	100	200	300	400	500
絶対誤差	m		2.861 ⁻⁵	1.831 ⁻⁴	1.382 ⁻³	4.364 ⁻³	1.028 ⁻²	1.932 ⁻²
	f		1.425 ⁻⁵	9.268 ⁻⁵	6.725 ⁻⁴	2.194 ⁻³	5.010 ⁻³	9.580 ⁻³
相対誤差	m		4.328 ⁻⁶	1.477 ⁻⁵	5.417 ⁻⁵	1.195 ⁻⁴	2.023 ⁻⁴	3.148 ⁻⁴
	f		3.059 ⁻⁶	1.029 ⁻⁵	3.766 ⁻⁵	8.274 ⁻⁵	1.409 ⁻⁴	2.176 ⁻⁴
$\log K - \log N_c$			5.850	5.407	4.959	4.696	4.510	4.365
$\log K - \log M_c$			4.721	4.124	3.524	3.172	2.923	2.729
$-\log m$			5.364	4.831	4.266	3.923	3.694	3.502
$-\log f$			5.514	4.987	4.424	4.082	3.851	3.662

表 4-2

 B_N ; Double

		N	50	100	200	300	400	500
絶対誤差	m		6.939 ⁻¹⁶	6.009 ⁻¹⁵	3.633 ⁻¹⁴	1.301 ⁻¹³	3.078 ⁻¹³	5.853 ⁻¹³
	f		3.584 ⁻¹⁶	2.944 ⁻¹⁵	1.880 ⁻¹⁴	6.461 ⁻¹⁴	1.496 ⁻¹³	2.918 ⁻¹³
相対誤差	m		1.168 ⁻¹⁶	4.540 ⁻¹⁶	1.565 ⁻¹⁵	3.492 ⁻¹⁵	6.068 ⁻¹⁵	9.493 ⁻¹⁵
	f		7.864 ⁻¹⁷	3.223 ⁻¹⁶	1.071 ⁻¹⁵	2.431 ⁻¹⁵	4.215 ⁻¹⁵	6.608 ⁻¹⁵
$\log K - \log N_c$			16.39	15.94	15.50	15.23	15.05	14.90
$\log K - \log M_c$			15.26	14.66	14.06	13.71	13.46	13.27
$-\log m$			15.93	15.34	14.81	14.46	14.22	14.02
$-\log f$			16.10	15.49	14.97	14.61	14.38	14.18

表 4-3

 B_N^2 ; Single

		N	50	100	200	300	400	500
絶対誤差	m		9.321 ⁻¹	2.034 ⁺¹	2.244 ⁺⁴	2.517 ⁺⁵	9.666 ⁺⁵	2.238 ⁺⁶
	f		4.606 ⁻¹	1.030 ⁺¹	1.114 ⁺⁴	1.258 ⁺⁵	4.848 ⁺⁵	1.116 ⁺⁶
相対誤差	m		3.746 ⁻⁴	1.660 ⁻³	1.563 ⁻¹	5.275 ⁻¹	8.467 ⁻¹	9.712 ⁻¹
	f		3.239 ⁻⁴	1.104 ⁻³	1.332 ⁻¹	4.498 ⁻¹	7.341 ⁻¹	8.627 ⁻¹
$\log K - \log N_c$			2.915	1.876	8.306 ⁻¹	2.168 ⁻¹	-2.192 ⁻¹	-5.576 ⁻¹
$\log K - \log M_c$			1.908	7.169 ⁻¹	-4.807 ⁻¹	1.183	-1.682	-2.069
$-\log m$			3.426	2.780	8.061 ⁻¹	2.778 ⁻¹	7.228 ⁻²	1.269 ⁻²
$-\log f$			3.490	2.957	8.757 ⁻¹	3.470 ⁻¹	1.342 ⁻¹	6.415 ⁻²

表 4-4

 B_N^2 ; Double

		N	50	100	200	300	400	500
絶対誤差	m		8.552 ⁻¹¹	5.349 ⁻⁹	6.394 ⁻⁷	7.483 ⁻⁶	5.926 ⁻⁵	3.288 ⁻⁴
	f		4.362 ⁻¹¹	2.725 ⁻⁹	3.195 ⁻⁷	3.758 ⁻⁶	3.004 ⁻⁵	1.650 ⁻⁴
相対誤差	m		3.814 ⁻¹⁴	3.236 ⁻¹³	4.590 ⁻¹²	1.626 ⁻¹¹	5.550 ⁻¹¹	1.539 ⁻¹⁰
	f		3.171 ⁻¹⁴	2.598 ⁻¹³	3.836 ⁻¹²	1.356 ⁻¹¹	4.610 ⁻¹¹	1.290 ⁻¹⁰
log $K - \log N_c$			13.45	12.41	11.37	10.75	10.32	9.978
log $K - \log M_c$			12.44	11.25	10.06	9.353	8.854	8.467
- log m			13.42	12.49	11.34	10.79	10.26	9.813
- log f			13.50	12.59	11.42	10.87	10.34	9.889

表 4-5

BNDTIS, BNSLIS
 $B_n \otimes O_{10}^{(1,D)}$; Single $N=10n$

		N	50	100	200	300	400	500
絶対誤差	m		2.772 ⁻⁵	4.840 ⁻⁵	6.669 ⁻⁴	2.659 ⁻³	5.888 ⁻³	1.156 ⁻²
	f		4.024 ⁻⁶	6.612 ⁻⁶	9.904 ⁻⁵	4.115 ⁻⁴	1.083 ⁻³	1.966 ⁻³
相対誤差	m		5.776 ⁻⁵	2.007 ⁻⁴	2.459 ⁻⁴	8.773 ⁻⁴	2.031 ⁻³	2.787 ⁻³
	f		7.162 ⁻⁶	1.857 ⁻⁵	4.724 ⁻⁵	1.480 ⁻⁴	3.297 ⁻⁴	4.846 ⁻⁴
log $K - \log N_c$			5.030	4.656	4.244	3.993	3.812	3.671
log $K - \log M_c$			3.790	3.229	2.644	2.299	2.052	1.860
- log m			4.238	3.697	3.609	3.057	2.692	2.555
- log f			5.145	4.731	4.326	3.830	3.482	3.315

表 4-6

BNDTBS, BNSLBS
 $B_n \otimes O_{10}^{(1,d)}$; Single $N=10n$

		N	50	100	200	300	400	500
絶対誤差	m		1.953 ⁻⁵	2.009 ⁻⁴	1.957 ⁻³	2.626 ⁻³	8.378 ⁻³	1.902 ⁻²
	f		3.185 ⁻⁶	1.878 ⁻⁵	1.706 ⁻⁴	2.464 ⁻⁴	8.001 ⁻⁴	1.622 ⁻³
相対誤差	m		9.951 ⁻⁵	7.632 ⁻⁴	1.381 ⁻²	4.521 ⁻²	1.710 ⁻¹	1.796 ⁻¹
	f		1.057 ⁻⁵	3.974 ⁻⁵	4.049 ⁻⁴	7.317 ⁻⁴	2.417 ⁻³	2.986 ⁻³
log $K - \log N_c$			5.030	4.656	4.244	3.993	3.812	3.671
log $K - \log M_c$			3.790	3.229	2.644	2.299	2.052	1.860
- log m			4.002	3.117	1.860	1.345	7.671 ⁻¹	7.457 ⁻¹
- log f			4.976	4.401	3.393	3.136	2.617	2.525

表 4-7

BNDTIS, BNSLIS
 $B_n \otimes O_{10}^{(2,D)}$; Single $N=10n$

		N	50	100	200	300	400	500
絶対誤差	m		5.588 -9	3.725 -8	1.460 -7	7.157 -7	1.371 -6	2.079 -6
	f		8.544 -10	3.329 -9	1.716 -8	6.566 -8	1.281 -7	2.147 -7
相対誤差	m		7.377 -6	1.756 -5	2.880 -5	4.487 -5	6.686 -5	9.705 -5
	f		7.051 -7	1.826 -6	4.297 -6	8.376 -6	1.336 -5	1.838 -5
log K - log N_c			6.821	6.447	6.035	5.785	5.604	5.463
log K - log M_c			5.062	4.501	3.917	3.571	3.324	3.133
- log m			5.132	4.755	4.541	4.348	4.175	4.013
- log f			6.152	5.739	5.367	5.077	4.874	4.736

表 4-8

BNDTBS, BNSLBS
 $B_n \otimes O_{10}^{(2,D)}$; Single $N=10n$

		N	50	100	200	300	400	500
絶対誤差	m		4.133 -9	2.747 -8	6.910 -7	1.566 -6	5.696 -6	1.784 -5
	f		6.109 -10	3.259 -9	4.607 -8	1.276 -7	4.254 -7	8.788 -7
相対誤差	m		6.905 -6	4.449 -5	2.843 -4	5.079 -3	1.068 -2	2.924 -2
	f		5.665 -7	2.698 -6	1.579 -5	1.172 -4	2.535 -4	5.458 -4
log K - log N_c			6.821	6.447	6.035	5.785	5.604	5.463
log K - log M_c			5.062	4.501	3.917	3.571	3.324	3.133
- log m			5.161	4.352	3.546	2.294	1.971	1.534
- log f			6.247	5.569	4.802	3.931	3.596	3.263

逆行列の計算に要する時間を、BNSLIS, BNSLBSの仮引数NRをNとした場合について表4-9, 10に与える。テスト行列 P_N は

$$(P_N)_{ij} = \begin{cases} 2 & 1 \leq i = j \leq N \\ 1 & 1 \leq i \neq j \leq N \end{cases}$$

で与えられる。表4-9の見方をBNDTISの $N=100$ について説明すれば、左端がテスト行列 $B_N \otimes P_{10}$ に対する時間で、このテスト行列では行交換が全く行われない。その上段が単精度ルーチンBNDTISの数値で下段が倍精度ルーチンBNDTIDの数値である。中央がテスト行列 $B_N \times O_{10}^{(2,D)}$ に対する同様の数値で、右端がその場合の行交換の回数である。これらのテスト行列の帯巾 $M1 + M2 + 1$ は共に39であって、前者は行交換が全くない行列である。表4-10の見方も同様で、帯巾が39と79

である。行交換のない場合、LU-分解に要する計算量のオーダーは $N \times M1 \times M2$ であり、解をN個求める計算量のオーダーは $N^2 \times (M1 + M2)$ である。表4-10を見れば、LU-分解に要する時間が、 $M1 = M2 = 19$ のときにはNの1乗に依存しているのがよくわかる。 $M1 = M2 = 39$ のときにはNの1乗より大きいオーダーでふえている。又、この2つのテスト行列に関する限り、帯巾が2倍になれば時間の方は大略4倍になっている。一方解を求めるのに要する時間は、帯巾を固定した場合Nの2乗より大きいオーダーでふえている。又、次数Nを固定した場合、帯巾が2倍になれば時間の方もほぼ2倍になっている。

固有値関係のテストには

$$A = (a_{ij}) \quad a_{3i-2, 3i-2} = a_{3i-1, 3i-1} = a_{3i, 3i} = 11 - i$$

表 4-9

テスト行列 左 $B_n \otimes P_{10}$
右 $B_n \otimes O_{10}^{(2,d)}$ $N=10n$
単位 (sec)

N	BNDTIS			BNDTBS			BNSLIS		BNSLBS		
	単	倍		単	倍		単	倍	単	倍	
100	単	0.14	0.14	40	0.14	0.15	186	0.84	0.83	0.84	0.90
	倍	0.14	0.15		0.14	0.16		1.12	1.12	1.13	1.18
200	単	0.28	0.29	80	0.29	0.31	406	3.82	3.87	3.88	4.09
	倍	0.30	0.30		0.30	0.33		5.95	6.04	5.97	6.37
300	単	0.43	0.45	120	0.43	0.47	666	10.7	10.9	10.8	11.4
	倍	0.46	0.47		0.47	0.50		13.6	13.8	13.6	14.7
400	単	0.58	0.60	160	0.59	0.62	934	21.2	21.6	21.4	22.7
	倍	0.61	0.62		0.63	0.63		24.5	24.8	24.4	26.4

表 4-10

テスト行列 左 $B_n \times P_{10}$ $N=10n$
右 $B_m \times P_{20}$ $N=20m$
単位 (sec)

N	BNDTIS			BNDTBS			BNSLIS		BNSLBS		
	単	倍		単	倍		単	倍	単	倍	
100	単	0.14	0.49		0.14	0.44		0.84	1.40	0.84	1.45
	倍	0.14	0.54		0.14	0.49		1.12	1.63	1.13	1.69
200	単	0.28	1.10		0.29	0.99		3.82	6.87	3.88	7.07
	倍	0.30	1.21		0.30	1.13		5.95	7.89	5.97	8.23
300	単	0.43	1.68		0.43	1.52		10.7	20.4	10.8	20.8
	倍	0.46	1.88		0.47	1.73		13.6	23.2	13.6	23.6
400	単	0.58	2.30		0.59	2.04		21.2	41.2	21.4	41.8
	倍	0.61	3.11		0.63	2.55		24.5	—	24.4	—

$$1 \leq i \leq 9$$

$$a_{1,2} = a_{2,1} = a_{1,3} = a_{3,1} = 1$$

$$a_{i,i+3} = a_{i+3,i} = 1 \quad 1 \leq i \leq 27$$

$$a_{i,j} = 0 \quad \text{その他}$$

$$B = (b_{ij}) \quad b_{i,i} = 6 \quad 2 \leq i \leq 10 \quad b_{1,1} = b_{11,11} = 5$$

$$b_{i,i+1} = b_{i+1,i} = 3 \quad 2 \leq i \leq 9$$

$$b_{1,2} = b_{2,1} = b_{10,11} = b_{11,10} = 2$$

$$b_{i,i+2} = b_{i+2,i} = 1 \quad 1 \leq i \leq 9$$

$$b_{i,i+3} = b_{i+3,i} = 1 \quad 1 \leq i \leq 8$$

$$b_{i,j} = 0 \quad \text{その他}$$

$$C = \begin{pmatrix} D-I & & & \\ -I & & & O \\ & & & \\ O & & & -I \\ & & & -I & D \end{pmatrix}$$

$$D = \begin{pmatrix} 4-1 & & & \\ -1 & & & O \\ & & & \\ O & & & -1 \\ & & & -1 & 4 \end{pmatrix}$$

の3種類の行列 A, B, C を用いた。ここで、 A は、30次、 B は11次、 D は9次、 C は81次、 I は9次単位行列である。 A は4.999と5.0の間に3個の固有値がある。 B は、2重固有値4.0の他は十分に分離された固有値のみを有する。 C は、8個の単純固有値と32対の2重固有値及び1個の9重固有値を有し、それらは

$$\lambda_{i,j} = 4 - 2(\cos i\pi/10 + \cos j\pi/10) \quad 1 \leq i, j \leq 9$$

で与えられる。(単純固有値は $i=j \neq 5$ に対応し、9重固有値は $i+j=10$ に対応する)

BNDTBS による固有値の数の勘定がどの程度であるかを A, B, C をテスト行列にした場合の結果を次に示す。(λ は真の固有値、 R は近似固有値、 NE は R より大きい固有値の数) まず、 A の場合、

λ	R	NE
4.9999		15

4.99983 ……		
	4.9998	16
4.99978 ……		
	4.9997	17
4.99969 ……		
	4.9996	18

という十分な結果が得られる。Bの2重固有値の付近の場合には

λ	R	NE
	4.0000001	6
4.0	4.0	7
	3.9999999	8

という結果が得られ、単精度としては十分である。Cについても同程度の結果が得られた。(表4-11参照)

inverse iteration法による固有ベクトルの計算及び固有値の改良結果について次に示す。行列Aの正確な固有ベクトルはわからないので、倍精度ルーチンで計算した値を正確な値として、それと単精度ルーチンとの結果を比較した。結果の整理には、正確な固有ベクトルをx、各ルーチンで計算した固有ベクトルをyとして、それぞれのユークリッドノルムが1になるように正規化($\|x\| = \|y\| = 1$)した後、

$$m = \|x - y\|_{\infty} \quad \|x\|_{\infty} = \max |x_i|$$

$$f = \|x - y\| \quad \|x\| = (\sum x_i^2)^{1/2}$$

の2つの値を求めた。今回用いたテスト行列の固有ベクトルはすべて実であり、単純固有値に対応する固有ベクトルは一意に定まるので、xとyのなす角は

$$\cos \theta = (x, y) = \sum x_i y_i$$

で与えられる。これは又

$$1 - \cos \theta = 1/2 f^2$$

とも表わされ、fが小さいときには $\theta = f$ となる。

表4-12に行列Aの近接固有値の付近におけるテスト結果を、表4-13、14に行列Bの大きい方と小さい方の固有値の付近におけるテスト結果を与える。Aをテスト行列とした場合のBSIVTSによる固有値の改良が良くないのは、左固有ベクトルの計算誤差によるものと思われる。Bをテスト行列にした場合には、それぞれの精度

の範囲で正確な値と一致した。次に、最初に与える固有値の近似度の影響を行列Bの単純固有値6の付近でテストしたのが表4-15、16である(テストはBSIVTSについてのみ行った)。表4-15の値は、イタレイションの上限値を100として1度BSIVTSをコールした場合の結果である。固有値が十分分離されていれば、かなり雑な近似値でも正確な固有値に改良されており、又固有ベクトルの精度にも変わりがない。しかし、このような方法ではイタレイションの回数が多くなるので、最初に小さい上限値を与えて固有値の改良を行ってから、もう1度コールするのがよい。表4-16は、最初に上限値5を与えて1度コールし、ついで改良された固有値を2度目にコールするときの近似固有値にとり又上限値を100にした場合の結果である。精度にはさほどの差はない(5.2の場合を除く)が、イタレイション回数は格段に減少している。代わりに、LU-分解が1度多くなったことを忘れてはいけない。なお、ここで注意を要するのは、最初の近似値が悪かったり、近接固有値があったりした場合には、このような手段を講じても予定している固有値に収束しなかったり、1度目にコールする上限値によって異なる固有値に収束したりすることがある。例えば、行列Aで近似固有値として5.0を与え、上限値を100としてBSIVTSをコールすれば12回のイタレイションで $\lambda = 4.9998326$ に収束するが、上限値を1として1度コールしてからでは9回のイタレイションで $\lambda = 4.9997824$ に収束する。又行列Bで近似固有値として5.2をとった場合、1度目のコールの上限値が1の場合には $\lambda = 4.4066499$ に収束し、1度目の上限値が3の場合には $\lambda = 6.0$ に収束した。このようではあるが、固有値が十分分離されているか、最初に与える近似値が良いものであるか、inverse iteration法は効果的な方法である。さらに多重固有値の場合を行列Bの2重固有値 $\lambda = 4$ についてテストした結果を表4-17に与える。今の場合Bが対称行列なので1次独立な固有ベクトルが2つ存在する。近似固有値として4.001と3.999をとって計算した結果を見ると、十分2つの1次独立な固有ベクトルを与えているのがわかる。(表中、 e_1 と e_2 は1次独立な固有ベクトルの例)

表 4-11

λ		2.38196601 …			4.0			5.0	
R	2.3819661		2.3819660	4.0000001	4.0	3.9999999	5.0000001	5.0	4.9999999
NE	62		64	36	45	45	24	25	26

表 4-12

NU = 100

R	インプット	4. 9997	4. 9998	4. 9999
	アウトプット	4. 9996896	4. 9997824	4. 9998326
λ		4. 99968956 ...	4. 99978247 ...	4. 99983258 ...
m		7. 477 -4	1. 378 -3	1. 840 -3
f		1. 562 -3	3. 374 -3	4. 048 -3
NE		2	9	3

BSIVTS

R	インプット	4. 9997	4. 9998	4. 9999
	アウトプット	4. 9996896	4. 9997822	4. 9998374
λ		4. 99968956 ...	4. 99978247 ...	4. 99983258 ...
m		4. 273 -2 7. 213 -4	8. 547 -1 1. 338 -3	7. 192 -1 2. 591 -3
f		7. 381 -2 1. 482 -3	1. 716 3. 279 -3	1. 364 6. 115 -3
NE		9 2	46 9	* 100 2

BUIVTS

左：左固有ベクトル
右：右固有ベクトル

* 収束しなかったので実際のアウトプットは 101

表 4-13

NU = 100

R	14. 94		12. 20		1. 804		0. 5223	
m	7. 451 -9		1. 490 -8		1. 490 -8		7. 451 -9	
f	1. 624 -8		3. 469 -8		3. 272 -8		1. 086 -8	
NE	2		2		1		1	
m	5. 148 -3	7. 451 -9	4. 371 -3	1. 490 -8	6. 140 -4	1. 490 -8	1. 025 -4	7. 451 -9
f	8. 674 -3	1. 624 -8	6. 053 -3	3. 469 -8	7. 029 -4	3. 272 -8	1. 627 -4	1. 086 -8
NE	2	2	3	2	2	1	1	1

上：BSIVTS

左：左固有ベクトル

下：BUIVTS

右：右固有ベクトル

表 4-14

NU = 100

R	14. 94		12. 20		1. 804		0. 5223	
m	4. 337 -19		2. 168 -19		2. 168 -19		6. 505 -19	
f	8. 691 -19		3. 098 -19		3. 559 -19		9. 327 -19	
NE	5		6		4		3	
m	5. 148 -3	4. 337 -19	4. 371 -3	2. 168 -19	6. 139 -4	2. 168 -19	1. 023 -4	6. 505 -19
f	8. 674 -3	8. 691 -19	6. 053 -3	3. 098 -19	7. 027 -4	3. 559 -19	1. 622 -4	9. 327 -19
NE	7	5	7	6	5	4	4	3

上：BSIVTD

左：左固有ベクトル

下：BUIVTD

右：右固有ベクトル

表 4-15

NU = 100

R	インプット	7.5	7.4	7.3	7.2	6.1	6.0	5.9	5.3	5.2	5.1
	アウトプット	λ_3	6.052	6.0	6.0	6.0	6.0	6.0	6.0	5.307	λ_5
	m	—	5.856 ⁻²	4.256 ⁻⁸	2.980 ⁻⁸	2.980 ⁻⁸	7.451 ⁻⁹	7.451 ⁻⁹	5.933 ⁻⁸	3.478 ⁻¹	—
	f	—	1.358 ⁻¹	8.509 ⁻⁸	6.542 ⁻⁸	4.629 ⁻⁸	1.325 ⁻⁸	1.288 ⁻⁸	1.310 ⁻⁷	7.045 ⁻¹	—
	NE	*100	*100	*100	54	5	1	6	60	*100	60

$\lambda_3 = 6 + 2\sqrt{2} = 8.828 \dots\dots$

$\lambda_5 = 6 - 5/2\sqrt{2} - \sqrt{3} + 3/2\sqrt{6} = 4.406 \dots\dots$

* 収束しなかったので実際のアウトプットは101

表 4-16

1回目 NU = 5

2回目 NU = 100

R	インプット	7.3		7.2		5.3		5.2	
	アウトプット	6.355	6.0	6.079	6.0	5.970	6.0	5.592	6.0
	m	1.668 ⁻¹	2.235 ⁻⁸	7.263 ⁻²	7.950 ⁻⁹	7.550 ⁻²	7.451 ⁻⁹	3.217 ⁻¹	4.470 ⁻⁸
	f	3.603 ⁻¹	5.359 ⁻⁸	1.681 ⁻¹	8.676 ⁻⁹	1.309 ⁻¹	2.157 ⁻⁸	5.016 ⁻¹	1.076 ⁻⁷
	NE	*5	9	*5	5	*5	4	*5	14

* 1回目のコールではすべて収束していない。

表 4-17

4.001	3.999	e1	e2
5.6565224 ⁻¹	-9.9999864 ⁻¹	1.0	1.0
2.1717387 ⁻¹	9.2776169 ⁻¹	0.0	-1.0
-7.8282611 ⁻¹	7.2236825 ⁻²	-1.0	0.0
-2.1717384 ⁻¹	-9.2776144 ⁻¹	0.0	1.0
1.0	8.5552485 ⁻¹	1.0	-1.0
-9.1031936 ⁻⁸	-6.4743071 ⁻⁷	0.0	0.0
-9.9999996 ⁻¹	-8.5552448 ⁻¹	-1.0	1.0
2.1717400 ⁻¹	9.2776255 ⁻¹	0.0	-1.0
7.8282599 ⁻¹	-7.2237814 ⁻²	1.0	0.0
-2.1717397 ⁻¹	-9.2776233 ⁻¹	0.0	1.0
-5.6565204 ⁻¹	1.0	-1.0	-1.0

5 章 結 語

最も簡単な問題であると思われる，1 次方程式系

$$Ax = b$$

を計算機で解くだけでも色々な解法を考慮する必要がある。まず簡単な経歴から云うと，初期の頃にここで採用したいわゆる直接法の系統に属する解法の誤差評価としてノイマンが与えたものは甚だ大きかった。その結果直接法では解が得られないかもしれないという危惧から，ガウス・ザイデル法等の逐次法が主として用いられた。その後ウィルキンソンがその独自の誤差評価法により，直接法でも十分な精度が得られることを示すや，現今はたいいていの計算ライブラリーにこの LU 分解とは本質的に同等であるものが存在する。そしていかなる場合にもこれによって解くことが可能と暗黙のうちに考えられ，自分の解こうとする問題の性質を考慮することがおろそかになってきている。それが大いなる誤りであることは，ウィルキンソンの評価そのものを見ても明らかであるし，或は又報告のテスト結果を見ても理解できる。さらに云えば，いかなる解法も解くべき問題の性質を考慮して取捨する，というのが計算機でなんらかの問題を解こうとする者の基本的姿勢でなければならない。

しからば以上でこの問題が完結したかといえれば決してそうではないのである。まずその理由の第一に挙げねばならないものとして，解法-ソフトウェアというものがそれが適用される計算機-ハードウェアに依存して発達するということがある。そして今や科学技術計算の分野ではその特殊性に応じたハードウェアとしてベクトルマシンと呼ばれるもの（パイプライン型，プロセッサレイ型）が普通化してきている。そうすればおのずと

今までの解法が，精度・速度さらには記憶容量などの点から再考されねばならなくなる。次に考える必要のある事柄は，行列 A が何らかの特殊性を有する場合である。その一つが帯状ということであるがそれ以外にも，正値対称であるとか 3 重対角行列である（いわゆる 3 項方程式），優対角行列である，帯状ではあるがその帯内でも非零要素が非常に少ない，いわゆるスパーズ行列である，さらには非常に巨大な大きさに到底インコアでは不可能である，等といった個々の場合にどう処理すればよいか，それとハードウェアの進歩とをどう調和させるかが今後共大きな課題である。

もっと身近な事柄としては，すでにあるルーチンの性能向上，維持管理がある。本報告中のものの中にも，帯状行列の場合の $Lux = b$ を解くのに要する時間が計算量から推定されるより多く必要となることや，inverse iteration 法によって左固有ベクトルを計算した場合，右固有ベクトルの場合程精度が良くないこと等について調べる必要がある。又，非対称帯状行列についてもテストしてみる必要がある。

以上のような点を考えれば，今後共この方面の研究が引き続き系統的に行われる必要性のあることがわかる。

参 考 文 献

- 1) 福田正大ほか；1 次方程式系の解法 I，航技研資料 TM-277 (1975.5)
- 2) J. H. Wilkinson; The Algebraic Eigenvalue Problem, London, Oxford Univ. Press (1965).
- 3) J. H. Wilkinson; Rounding Errors in Algebraic Processes, London, Her Majesty's Stationery Office, Englewood Cliffs, Prentice-Hall (1963).

航空宇宙技術研究所資料348号

昭和53年4月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話武蔵野三鷹(0422)47-5911(大代表)〒182
印刷所 株式会社 共 進
東京都杉並区久我山4-1-7(羽田ビル)
