第1章　連立1次方程式の解法—直 接 法

連立 1 次方程式 (system of linear equations, simultaneous linear equations) の数値解法はいろいろあるが，普通用いられているものは直接法と反復法である．直接法は連立 1 次方程式を行列計算によって直接解くものである．一方 反復法は，楕円型微分方程式の境界値問題を解く際にできるような大形疎行列の連立 1 次方程式に適用されるもので，なんらかの予測値を与え，予測値の修正計算を反復し，その収束解を求めるものである．最近コンピュータが高速大容量化しているとはいえ，大量の連立 1 次方程式を繰返し解かなければならない場合には計算量の少ない解法を選ぶべきで，それにはまず各解法の特徴を知ることが必要である．直接法は通常 ライブラリ・サブルーチンを利用できるが，反復法のプログラムは自分で作らなければならない．

連立 1 次方程式

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{1.1}$$

は１つの式にまとめて

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad (i = 1, 2, \ldots, n) \tag{1.1a}$$

のように，あるいは行列を用い

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \tag{1.1b}$$

のように書くことができる．また簡単に

$$Ax = b \tag{1.1c}$$

のように書くこともある．ただし A は係数行列，x は未知変数のベクトル，b は定数ベクトルである．

直接法 (direct method) にはいろいろの解法があり，係数行列 A が帯行列の場合に適するもの，同じ A を持ち定数ベクトル b の異なるいくつかの連立 1 次方程式を解く場合に適するものがある．本章には次の４つの典型的な解法について述べる．

A ＼ b	1つ	いくつか
一般の行列	Gauss-Jordan 法	逆行列 A^{-1}
帯行列	Gauss 消去法	Crout 法

1.1 行列演算

解法の説明に入る前に，行列とその演算について簡潔に述べる．行数が I，列数が J の $I\times J$ 行列は

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \vdots & \vdots & & \vdots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix} \equiv [a_{ij}] \equiv A$$

のように表わされる．

行列演算 [加法]： A と B がともに $I\times J$ 行列ならば，1 次結合を取ることができる．

$$\lambda A + \mu B = C \tag{1.2}$$

ただし λ, μ はスカラー量，C は $I\times J$ 行列でその要素は $c_{ij} = \lambda a_{ij} + \mu b_{ij}$ である．

[乗法]： A が $I\times J$ 行列，B が $J\times K$ 行列で，A の列数と B の行数が等しければ乗算可能である．

$$AB = C \tag{1.3}$$

C は $I\times K$ 行列でその要素は $c_{ik} = \sum_{j=1}^{J} a_{ij}b_{jk}$ である．

$1\times J$ 行列は J 次の行ベクトル (row vector)，$I\times 1$ 行列は I 次の列ベクトル (column vector)，行数と列数の等しい行列は正方行列 (square matrix) と呼ばれる．正方行列の対角線上に並ぶ要素 a_{ii} は主対角要素 (diagonal elements) と呼ばれる．正方行列 $[a_{ij}]$ において $j < i$ の要素がすべてゼロの行列は上三角行列 (upper triangular matrix)，$j > i$ の要素がすべてゼロの行列は下三角行列 (lower triangular matrix)，主対角要素 a_{ii} 以外の要素がすべてゼロの行列は対角行列 (diagonal matrix)，特に $a_{ii} = 1$ の対角行列は単位行列 (unit matrix, identity) と呼ばれる．ここでは単位行列を I で表すことにする．

$n\times n$ 行列 A の行列式 (determinant) は，

$$\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = |a_{ij}| = |A| = \det(A)$$

のように書かれる．$|A|$ はその要素 a_{ij} の高々n 次式である．

基本行演算： (i) $n\times n$ 行列 A の第 i 行に 0 でないスカラー量 λ をかけた行列を B とすれば，

$$B = E_i(\lambda)A \tag{1.4}$$

ただし $E_i(\lambda)$ は $n\times n$ 単位行列 I の i 番目の対角要素を λ 倍した行列で

$$E_{ik} = \begin{pmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \lambda & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{pmatrix} \; i\text{ 行} \tag{1.4a}$$

である．なお B の行列式は，$|B| = \lambda|A|$ となる．

(ii) 行列 A の第 i 行と第 k 行を入替えた行列を B とすれば

$$B = E_{ik}A \tag{1.5}$$

ただし E_{ik} は単位行列 I の i 行と k 行を入替えた行列で

$$E_{ik} = \begin{pmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & \cdots & 0 & \cdots & 1 & \cdots & \\ & & & \ddots & & & \\ & \cdots & 1 & \cdots & 0 & \cdots & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ i\text{ 行} \\ \\ k\text{ 行} \\ \\ \end{matrix} \tag{1.5a}$$

である．なお $|B| = -|A|$ である．

(iii) 行列 A の第 i 行に第 k 行を λ 倍したものを加えた行列を B とすれば

$$B = E_{ik}(\lambda)A \tag{1.6}$$

ただし E_{ik} は単位行列 I の i 行に k 行を λ 倍したものを加えた行列で

$$E_{ik}(\lambda) = \begin{pmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & \cdots & 1 & \cdots & \lambda & \cdots & \\ & & & \ddots & & & \\ & & \cdots & 1 & \cdots & & \\ & & & & \ddots & & \\ 0 & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ i\text{ 行} \\ \\ k\text{ 行} \\ \\ \end{matrix} \tag{1.6a}$$

である．なお $|B| = |A|$ である．

これらの基本行演算，式 (1.4)-(1.6) はより一般的な $n \times m$ 行列 A に対してもそのまま成立する．なおこのときも $E_i(\lambda)$，E_{ik}，$E_{ik}(\lambda)$ は $n \times n$ の正方行列であるが，B は $n \times m$ 行列になる．連立 1 次方程式の直接法は，係数行列 A にこれらの基本行演算を繰り返し行い，A を単位行列 I に持っていくものである．

$n \times n$ 行列 A から 1 つの行と 1 つの列を取去った行列の行列式は行列式 $|A|$ の $n-1$ 次の小行列式 (minor)，また行列 A から i 行と j 列を取去った小行列式に $(-1)^{i+j}$ をかけたものは余因子 (cofactor) と呼ばれる．要素 a_{ij} の余因子を A_{ij} で表せば

$$|A| = \sum_{j=1}^{n} a_{ij}A_{ij} = \sum_{i=1}^{n} a_{ij}A_{ij} \tag{1.7}$$

となる．行列式が 0 でない行列は正則行列といわれる．正則行列 A は逆行列 A^{-1} を持つ．

$$A^{-1}A = AA^{-1} = I$$

逆行列は次式で与えられる．

$$A^{-1} = \frac{1}{|A|}\mathrm{adj}(A) \tag{1.8}$$

ただし $\mathrm{adj}(A)$ は行列 A の随伴行列 (adjoint matrix) で次のように定義される．

$$\mathrm{adj}(A) = \begin{pmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \vdots & \vdots & & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{pmatrix} \tag{1.8a}$$

1.2 ガウス・ジョルダン法

一般に連立 1 次方程式は，その 1 つの式を定数倍しても，式の順番を替えても，あるいは 1 つの式に他の式を定数倍したものを加えても解は同じである．直接法はこれら 3 つの操作を繰り返し実行することによって連立 1 次方程式を解くものである．式 (1.1) の係数行列 A と定数ベクトル b を並べて書いた行列を

$$A^0 = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix}$$

とすれば，これら 3 つの操作の内，1 つの式を定数倍することは式 (1.4) の計算，2 つの式の順番を入替えることは式 (1.5) の計算，1 つの式に他の式を定数倍したものを加えることは式 (1.6) の計算を実行することに相当する．A の逆行列を A^{-1} とすれば

$$A^{-1}A^0 = \begin{pmatrix} 1 & 0 & \cdots & 0 & x_1 \\ 0 & 1 & \cdots & 0 & x_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & x_n \end{pmatrix}$$

であるから，行列 A^0 の A の部分を基本行演算を次々に実行することによって単位行列 I にもっていくことができれば，そのとき A^0 の b の部分は解 x になる．A を単位行列にもって行く方法すなわち直接法の計算手順はいろいろ考案されている．

はじめにガウス・ジョルダン法 (Gauss-Jordan reduction，掃出し法) について説明する．この解法の計算手順は，簡単な連立 1 次方程式

$$\begin{aligned} 2x_1 - 2x_2 + x_3 &= 3 \\ 3x_1 - 3x_2 + x_3 &= 6 \\ x_1 + x_2 - x_3 &= 2 \end{aligned} \tag{1.9}$$

に対しては，次の表 1.1 に示すようになる．第 1 段階 $k = 1$ では，A^0 の第 1 列を単位行列のものにする行演算が行われる．すなわち第 1 行を 1/2 倍し，第 2 行にこの第 1 行を -3 倍したものを加え，第 3 行に第 1 行を -1 倍したものを加える操作が行われる．$k = 2$ では，第 2 列の主対角要素 a_{22} がゼロになっているので，前半で第 2 行と第 3 行を入替え，後半で第 2 列が単位行列のものになるように操作する．$k = 3$ では，第 3 列が単位行列のものになるように操作する．この時点で A^0 は単位行列 I と解 x を並べたものになる．式 (1.9) の解は $x_1 = 1$, $x_2 = -2$, $x_3 = -3$ である．逆行列 A^{-1} は第 1 段階から第 3 段階までのすべての演算子の積で

$$A^{-1} = E_{23}(3/4)E_{13}(1/4)\cdots E_1(1/2) = \begin{pmatrix} 1 & -1/2 & 1/2 \\ 2 & -3/2 & 1/2 \\ 3 & -2 & 0 \end{pmatrix}$$

となる．また行列式 $|A|$ の値は，A を I に持って行く過程で，$k = 1$ で 2 で割算，$k = 2$ で行の入替と 2 で割算，$k = 3$ で $-1/2$ で割算しているので，$|A| = 2 \cdot (-2) \cdot (-1/2) = 2$ となる．

Gauss-Jordan 法の計算は，表 1.1 の計算手順を一般化したものである．その第 k 段階の計算は次のようになる．主対角要素の大きさ $|a_{kk}|$ が十分小さい正の数 ε よりも大きいときには下記の計算に直行し，小さ

表 1.1: Gauss-Jordan 法の計算手順

k	$A \to I$			$b \to x$	行列演算
0	2	−2	1	3	A^0
	3	−3	1	6	
	1	1	−1	2	
1	1	−1	1/2	3/2	$E_{31}(-1)E_{21}(-3)E_1(1/2)A^0 = A^1$
	0	0	−1/2	3/2	
	0	2	−3/2	1/2	
2	1	−1	1/2	3/2	
	0	2	−3/2	1/2	
	0	0	−1/2	3/2	
	1	0	−1/4	7/4	$E_{12}(1)E_2(1/2)E_{23}A^1 = A^2$
	0	1	−3/4	1/4	
	0	0	−1/2	3/2	
3	1	0	0	1	$E_{23}(3/4)E_{13}(1/4)E_3(-2)A^2$
	0	1	0	−2	
	0	0	1	−3	

いときにはまず表 1.1 の $k = 2$ 欄に示すように $|a_{ik}|$ $(i = k + 1, \ldots, n)$ の中から ε よりも大きいものを捜し行の入替えを行う．次に $j = k$ 列が単位行列のものになるように，次の (a) 式で $i = k$ 行の計算を行い，それから (b) 式で残りの $i \neq k$ 行の計算を行う[1]．

$$a_{ij} = a_{ij}/a_{ii} \qquad (i = k;\ j = k{+}1, \ldots, n, n{+}1) \tag{1.10a}$$

$$a_{ij} = a_{ij} - a_{ik}a_{kj} \qquad (i \neq k;\ j = k{+}1, \ldots, n, n{+}1) \tag{1.10b}$$

以下に Gauss-Jordan 法の FORTRAN 95/90 Free format source form のプログラムを示す．なお本書のプログラムはすべて簡単なコメント付きになっている．！以下の文はコメントで計算に直接関係のないものである．

```
!*********************************************************************
!  Solution of system of linear equations by Gauss-Jordan reduction
!*********************************************************************
SUBROUTINE GAUJOR(a,wa,n,n1,det,e)
DIMENSION  a(n,n1),wa(n1)
det=1.
e=MAX(e,1.E-5)
DO k=1,n
  w=a(k,k); IF(ABS(w)>e)           GOTO 11        !|a(kk)|>e?
  l=k; 12 l=l+1; IF(l>n)STOP 5555                 !improper coef matrix
  w=a(l,k); IF(ABS(w)<=e)          GOTO 12
  FORALL(j=k:n1)                                  !exchange rows
    wa(j)=a(k,j); a(k,j)=a(l,j); a(l,j)=wa(j)
  ENDFORALL
  det=-det; 11 det=w*det                          !compute |A|
  FORALL(j=k+1:n1)a(k,j)=a(k,j)/w                 !compute eq(1.10a)
  cycle_1: DO i=1,n; IF(i==k)CYCLE cycle_1
    FORALL(j=k+1:n1)a(i,j)=a(i,j)-a(i,k)*a(k,j)   !compute eq(1.10b)
  ENDDO cycle_1
ENDDO
END
```

[1] 式 (1.10) はプログラムに関する式で，右辺は計算前，左辺は計算後の変数の値を意味する．プログラムの変数は記憶場所を示し，その値は計算の過程で変わる．

このプログラムについて簡単に説明する．サブルーチン GAUJOR は，引数の配列 a に係数行列 A と定数ベクトル b の値を並べて入力し，n= n，n1= n+1，e= ε と置いて計算を実行すれば，a(i,n1) の b を入力したところに解 x，det に行列式 $|A|$ の値が出力されるものである．なおここでは $j = k$ 列の計算を省いているので，A は単位行列ではなく意味のないものになる．計算の過程で，主対角要素 a_{kk} の絶対値が ε よりも小さいときには，この値が大きくなるまで次々に行の入替えが行われるが，最後まで行っても大きくならないときには計算は途中で STOP し，5555 が出力される．このような事態は，$|A|$ の値がゼロまたはそれに近く，解こうとしている連立 1 次方程式が不適切なときに起きる．計算を続けても意味のある結果は得られない．MAIN PROGRAM，アルゴリズム，更には問題の立て方に立ち返ってその原因を探る必要があろう．

1.3 消去法による逆行列 A^{-1} の計算

同じ係数行列 A を持つ 2 つの連立 1 次方程式を解く問題を考える．その定数ベクトルを b^1, b^2 とする．この問題では，前節の Gauss-Jordan 法のサブルーチン GAUJOR を用い，配列 A に A, b^1, b^2 を並べて入力し n1= n+2 と置けば，二つの解 x^1, x^2 を同時に求めることができる．同じ係数行列 A を持つ多くの連立 1 次方程式を解く場合，また定数ベクトル b の値が次々に更新される場合には，逆行列 (inverse matrix) A^{-1} をあらかじめ計算し，

$$x = A^{-1}b \tag{1.11}$$

から解 x を求める方法が便利である．逆行列 A^{-1} は，サブルーチン GAUJOR を用い計算することができる．すなわち配列 A に A と単位行列 I を並べて入力し，n1= $2n$ と置いて計算を実行すれば，I を入力したところに A^{-1} が出力される．しかしこの方法では実数の記憶容量は $2n^2$ 個必要になるので，通常はその半分の n^2 個で済ませる方法が用いられている．

今 $n \times 2n$ の配列 A において係数行列 A を入力する左半分を配列 L，単位行列 I を入力する右半分を配列 R とする．行の入替えがなければ，第 k 段階の計算を終わった時点で，配列 L の第 1 列から k 列までと配列 R の第 k+1 列から n 列までが単位行列のものになり，必要なデータは残りの部分に書き込まれる．したがって単位行列の部分を省いた $n \times n$ 配列の中にすべての必要なデータを記憶し，またその中ですべての計算を実行することができることになる．しかしながら行の入替えのある場合には簡単にはいかない．このことを上記の式 (1.9) の係数行列で見てみよう．表 1.2 は逆行列を求める計算手順を示したものである．この表の第 2 段階 $k = 2$ の前半は，主対角要素 $|a_{22}|$ を ε よりも大きくするための行の入替えである．この行の入替えによって配列 R の単位行列の部分も第 2 列と第 3 列が入替わることになる．また $k = 2$ の後半は L の第 2 列を単位行列のものにする計算で，R の必要なデータが第 2 列にではなく第 3 列に現れることになる．また第 3 段階では R の必要なデータが第 2 列に加わる．配列 L と R のデータを $n \times n$ の配列 A にまとめて記憶する場合には，第 k 段階の計算で R のデータは行の入替えにかかわらず A の空いている第 k 列に書き込まざるをえない．つまり行の入替えに伴って列も入替える必要がある．そのため計算が一通り終わった時点で入替えた列を元に戻してやる操作が必要になる．

Gauss-Jordan 法による逆行列の計算は，表 1.2 の計算手順を一般化したもので，その計算式は式 (1.10) と同様である．次に行列反転のプログラムを示す．

```
!*************************************************
!     Matrix inversion by Gauss-Jordan reduction
!*************************************************
SUBROUTINE MATINV(a,wa,iw,n,det,e)
DIMENSION  a(n,n),wa(n),iw(n)
```

表 1.2: Gauss-Jordan 法による逆行列の計算手順

k	L $(A \to I)$			R $(I \to A)$			A		
0	2	−2	1	1	0	0	2	−2	1
	3	−3	1	0	1	0	3	−3	1
	1	1	−1	0	0	1	1	1	−1
1	1	−1	1/2	1/2	0	0	1/2	−1	1/2
	0	0	−1/2	−3/2	1	0	−3/2	0	−1/2
	0	2	−3/2	−1/2	0	1	−1/2	2	−3/2
2	1	−1	1/2	1/2	0	0	1/2	−1	1/2
	0	2	−3/2	−1/2	0	1	−1/2	2	−3/2
	0	0	−1/2	−3/2	1	0	−3/2	0	−1/2
	1	0	−1/4	1/4	0	1/2	1/4	1/2	−1/4
	0	1	−3/4	−1/4	0	1/2	−1/4	1/2	−3/4
	0	0	−1/2	−3/2	1	0	−3/2	0	−1/2
3	1	0	0	1	−1/2	1/2	1	1/2	−1/2
	0	1	0	2	−3/2	1/2	2	1/2	−3/2
	0	0	1	3	−2	0	3	0	−2
4							1	−1/2	1/2
							2	−3/2	1/2
							3	−2	0

```
det=1.
e=MAX(e,1.E-5)
FORALL(k=1:n)iw(k)=k
DO k=1,n
  w=a(k,k); IF(ABS(w)>e)                GOTO 11   !|a(kk)|>e?
  l=k; 12 l=l+1; IF(l>n)STOP 5555                 !improper coef matrix
  w=a(l,k); IF(ABS(w)<=e)               GOTO 12
  FORALL(j=1:n)                                   !exchange rows
    wa(j)=a(k,j); a(k,j)=a(l,j); a(l,j)=wa(j)
  ENDFORALL
  iw0=iw(k); iw(k)=iw(l); iw(l)=iw0               !keep row numbers in iw
  det=-det; 11 det=w*det                          !compute |A|
  a(k,k)=1.
  FORALL(j=1:n)a(k,j)=a(k,j)/w                    !compute eq(1.10a)
  cycle_1: DO i=1,n; IF(i==k)CYCLE cycle_1
    w=a(i,k); a(i,k)=0.
    FORALL(j=1:n)a(i,j)=a(i,j)-w*a(k,j)           !compute eq(1.10b)
  ENDDO cycle_1
ENDDO
! exchange columns caused by exchange rows
cycle_2: DO j=1,n; IF(iw(j)==j)CYCLE cycle_2
  l=j; 21 l=l+1; IF(iw(l)/=j)           GOTO 21   !#1
  FORALL(i=1:n)
    wa(i)=a(i,j); a(i,j)=a(i,l); a(i,l)=wa(i)
  ENDFORALL
  iw(l)=iw(j)
ENDDO cycle_2
END
```

このプログラムは基本的に上記のサブルーチン GAUJOR と同じである．ただし iw は行の入替えを記録する配列で，始めに 1 から n までの整数を入れ，配列 a の行の入替えと同時に iw の要素も入替えている．#1 以降は a の入替えた列を元に戻す操作で，iw の要素を元に戻しながら同時に a の列も入替えている．

1.4 ガウス消去法

Gauss-Jordan 法が係数行列 A を直接 単位行列 I に変換するものであるのに対し，ガウス消去法 (Gaussian elimination) は A をいったん上三角行列 U に変換しそれから I に変換するものである．大きい連立 1 次方程式では，この前段の計算量は Gauss-Jordan 法の半分に近く，また後段の U の連立 1 次方程式を解く計算量も少ないので，ガウス消去法の計算量は Gauss-Jordan 法に比べかなり少なくなる．Gauss 消去法の計算手順は，前記の簡単な連立 1 次方程式 (1.9) に対しては表 1.3 のようになる．この表の左半分は $A \to U$ の変換で，その第 k 段階では k 行以降のみを Gauss-Jordan 法と同様に計算している．また右半分は $U \to I$ の変換で，$k = 3$ では $E_{23}(-a_{23})E_{13}(-a_{13})$ を，$k = 2$ では $E_{12}(-a_{12})$ を演算している．以下にはこれを一般化した Gauss 消去法のプログラムを示す．

表 1.3: Gauss 消去法の計算手順

k	$A \to U$			$b \to c$	k	$U \to I$			$c \to x$
	2	−2	1	3		1	−1	1/2	3/2
0	3	−3	1	6		0	1	−3/4	1/4
	1	1	−1	2		0	0	1	−3
	1	−1	1/2	3/2		1	−1	0	3
1	0	0	−1/2	3/2	3	0	1	0	−2
	0	2	−3/2	1/2		0	0	1	−3
	1	−1	1/2	3/2		1	0	0	1
2	0	2	−3/2	1/2	2	0	1	0	−2
	0	0	−1/2	3/2		0	0	1	−3
	1	−1	1/2	3/2					
	0	1	−3/4	1/4					
	0	0	−1/2	3/2					
	1	−1	1/2	3/2					
3	0	1	−3/4	1/4					
	0	0	1	−3					

```
!*****************************************************************
!  Solution of system of linear equations by Gaussian elimination
!*****************************************************************
SUBROUTINE GAUSS(a,wa,n,n1,det,e)
DIMENSION  a(n,n1),wa(n1)
det=1.
e=MAX(e,1.E-5)
cycle_1: DO k=1,n; l=k                                  !The first step computation
  w=a(k,k); IF(ABS(w)>e)                GOTO 11         !|a(kk)|>e?
  l=k; 12 l=l+1; IF(l>n)STOP 5555                       !improper coef matrix
  w=a(l,k); IF(ABS(w)<=e)               GOTO 12
  FORALL(j=k:n1)                                        !exchange rows
    wa(j)=a(k,j); a(k,j)=a(l,j); a(l,j)=wa(j)
  ENDFORALL
  det=-det; 11 det=w*det                                !compt |A|
  FORALL(j=k+1:n1)a(k,j)=a(k,j)/w                       !compt (1.10a)
  IF(k==n)CYCLE cycle_1
  FORALL(i=k+1:n,j=k+1:n1)a(i,j)=a(i,j)-a(i,k)*a(k,j)   !compt (1.10b)
ENDDO cycle_1
DO k=n,2,-1                                             !The second step computation
  FORALL(i=1:k-1)a(i,n1)=a(i,n1)-a(i,k)*a(k,n1)         !compt (1.10b)
ENDDO
END
```

このプログラムの説明は省略する．

帯行列 (band matrix) は，正方行列のゼロでない要素がすべて主対角要素まわりのある狭いバンド幅内に入るものをいう．言換えれば 帯行列は，$n\times n$ 正方行列 A の要素を a_{ij} とすれば，$j<i+m_1$ および $j>i+m_2$ の要素の値がすべてゼロになる行列である．ただし m_1 は 0 または負の整数，m_2 は 0 または正の整数，また $m_2-m_1+1=m$ はバンド幅でバンド内の左端から右端までの要素数である．帯行列の逆行列は一般に同じバンド幅の帯行列にはならない．また行の入替え時にもバンド幅は当然違ってくる．帯行列は，図 1.1 に示すように，$n\times m$ 配列 $\tilde{A}$ に記憶することができる．今簡単のため主対角要素の列番号を 0 とすれば，縮小行列 $\tilde{A}$ の要素 $\tilde{a}_{ij}$ と元の行列 A の要素 a_{ij} の関係は $\tilde{a}_{ij}=a_{i,j+i},\ a_{ij}=\tilde{a}_{i,j-i}$ である．特に主対角要素では $\tilde{a}_{i0}=a_{ii}$ である．帯行列の連立 1 次方程式が行の入替えなしに解ける場合には，Gauss 消去法で解けば計算の過程でゼロでない要素がバンド幅の外に出ることはなく，縮小行列 $\tilde{A}$ を用い計算に必要なメモリをバンド幅にもよるが大幅に節約することができる．

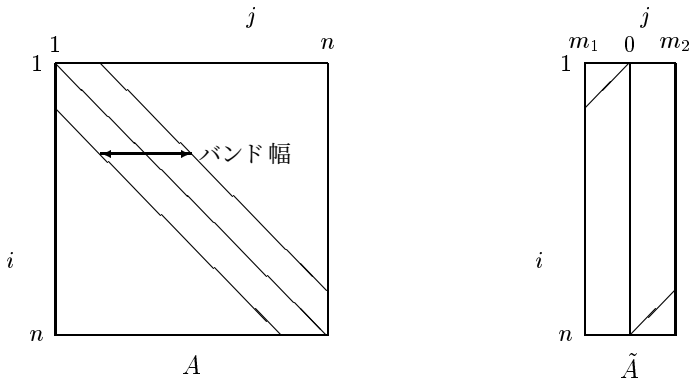

図 1.1: 帯行列 A とその縮小行列 $\tilde{A}$

縮小行列 $\tilde{A}$ に対する Gauss 消去法の計算式は，式 (1.10) を上の関係を用いて書換えた次式になる．

$$\tilde{a}_{ij}=\tilde{a}_{ij}/\tilde{a}_{i0} \qquad (j=1,\ldots,j_2), \qquad b_i=b_i/\tilde{a}_{i0} \qquad (i=k) \tag{1.12a}$$

$$\tilde{a}_{ij}=\tilde{a}_{ij}-\tilde{a}_{ij_0}\tilde{a}_{kj_k} \qquad (j_k=1,\ldots,j_2), \qquad b_i=b_i-\tilde{a}_{ij_0}b_k \qquad (i=k+1,\ldots,i_2) \tag{1.12b}$$

ただし $i_2=\min(n,\,k-m_1),\ j_2=\min(m_2,\,n-k),\ j=j_k+k-i,\ j_0=k-i$ である．この式を理解するには，行列 A と縮小行列 $\tilde{A}$ の図を描き，第 k 段階の計算の範囲をまず A に書入れ，これに対応する $\tilde{A}$ の範囲を調べる必要がある．式 (1.12a) では $\tilde{a}_{ij}$ の計算の範囲は $0\le j\le m_2$ でも良いが，ここでは必要最低限を取っている．式 (1.12b) で j は今計算している i 行上の列番号，j_k は k 行上の列番号で，j と j_k はもとの行列 A では同じ列上にある．また j_0 は $j_k=0$ の j である．$\tilde{a}_{ij}$ の計算の範囲は k 行上でみれば式 (1.12a) と同じになる．また i の範囲は $k+1\le i\le n$ であると同時に $m_1\le j_0=k-i$ でなければならない．結局第 k 段階で $\tilde{A}$ の計算すべき範囲は $\tilde{a}_{k1},\ \tilde{a}_{kj_2},\ \tilde{a}_{i_2,m_1+1}$ を頂点とする平行四辺形の中になる．

次に縮小行列 $\tilde{A}$ を用いる Gauss 消去法のプログラムを示す．

```
!************************************************************
!   Solution of system of linear equations with band matrix
!   by Gaussian elimination using a reduced array
!************************************************************
SUBROUTINE GAUSSB(at,b,n,m1,m2,det)
DIMENSION  at(n,m1:m2),b(n)
det=1.
cycle_1: DO k=1,n                                  !The first step computation
```

```
  det=at(k,0)*det
  b(k)=b(k)/at(k,0)                                    !compt (1.12a)
  IF(k==n)CYCLE cycle_1
  j2=MIN0(m2,n-k)
  FORALL(j=1:j2)at(k,j)=at(k,j)/at(k,0)                !compt (1.12a)
  i2=MIN0(n,k-m1)
  DO i=k+1,i2; j0=k-i
    b(i)=b(i)-at(i,j0)*b(k)                            !compt (1.12b)
    DO jk=1,j2; j=jk+k-i
      at(i,j)=at(i,j)-at(i,j0)*at(k,jk)                !compt (1.12b)
    ENDDO
  ENDDO
ENDDO cycle_1
DO i=n-1,1,-1                                          !The second step computation
  j2=MIN0(m2,n-i)
  DO j=1,j2; k=i+j
    b(i)=b(i)-at(i,j)*b(k)                             !compt (1.12b)
  ENDDO
ENDDO
END
```

at は縮小行列 $\tilde{A}$ の配列，b は係数ベクトル b を入力し解 x を出力する配列である．このプログラムは行の入替が必要な場合には使えない．

ここで 実際の計算で，帯行列 A の代わりに縮小行列 $\tilde{A}$ を用いた場合に，メモリをどの程度節約できるのかを見よう．例えば 2 次元の楕円型微分方程式の第 1 種境界値問題を 5 点差分式を用いて解く場合には，計算領域が 20×20 分割されていれば，未知変数の数は $19\times19=361$，a の要素数は $361\times361=130\ 321$，at の要素数は $361\times39=14\ 079$ になり，メモリは $1/9$ 程度になる．また 28×14 分割されるときには，未知変数の数は $27\times13=351$，a の要素数は $351\times351=123\ 201$，at の要素数は $351\times27=9\ 477$，メモリは $1/13$ 程度になる．またこのような問題ではデータの入力は配列 at の方が簡明になるという利点もある．

問 1　n 元連立 1 次方程式の係数行列が 3 重対角行列の場合に，この方程式を $n\times3$ 縮小配列を用いて解くプログラムを示せ．

ここでは主対角要素を $\tilde{a}_{k2}$ に取る．上記のプログラムは次のように簡単になる．

```
!**********************************************************************
!   Solution of system of linear equations with tri-diagonal matrix
!   by Gaussian elimination using reduced array
!**********************************************************************
SUBROUTINE GAUSS3(at,b,n)
DIMENSION  at(n,3),b(n)
DO k=1,n-1
  b(k)=b(k)/at(k,2)
  at(k,3)=at(k,3)/at(k,2)
  b(k+1)=b(k+1)-at(k+1,1)*b(k)
  at(k+1,2)=at(k+1,2)-at(k+1,1)*at(k,3)
ENDDO
b(n)=b(n)/at(n,2)
DO k=n-1,1,-1
  b(k)=b(k)-at(k,3)*b(k+1)
ENDDO
END
```

問 2　上記の問題で 3 重対角要素の他に 2 つの要素 a_{1n} と a_{n1} が加わった場合に，$n\times3$ 縮小配列の中で解くプログラムを示せ (このような問題は周期境界条件の課される場合に生じる)．

ここでは a_{1n} を空いている at(1,1) に，a_{n1} を at(n,3) に入力し，プログラムは上記の GAUSS3 に周期条件を加えたかたちに作る．$k = n$ のところがやや難解になるが，３重対角行列 A とその縮小行列 $\tilde{A}$ の図を基に考えれば容易に理解できよう．

```
!**********************************************************************
!  Solution of system of linear equations with modified tri-diagonal
!  matrix by periodic boundary condition using reduced array
!**********************************************************************
SUBROUTINE GAUSSP(at,b,n)
DIMENSION  at(n,3),b(n)
DO k=1,n-1
  b(k)   =b(k)   /at(k,2)
  at(k,3)=at(k,3)/at(k,2)
  at(k,1)=at(k,1)/at(k,2)                                  !period cond
  b(k+1)   =b(k+1)   -at(k+1,1)*b(k)
  at(k+1,2)=at(k+1,2)-at(k+1,1)*at(k,3)
  at(k+1,1)=         -at(k+1,1)*at(k,1)                    !period cond
  b(n)   =b(n)   -at(n,3)*b(k)                             !period cond
  at(n,2)=at(n,2)-at(n,3)*at(k,1)                          !period cond
  at(n,3)=       -at(n,3)*at(k,3)                          !period cond
ENDDO
b(n)=b(n)/(at(n,1)+at(n,2)+at(n,3))                        !period cond
DO k=n-1,1,-1
  b(k)=b(k)-at(k,1)*b(n)-at(k,3)*b(k+1)                    !period cond
ENDDO
END
```

1.5 クラウト法

クラウト法 (Crout method) は，連立 1 次方程式の係数行列 A を上下の三角行列 U と L の積

$$A = LU \tag{1.13}$$

に分解し，連立１次方程式 $LUx = b$ を 2 段階に分け，２つの三角行列の連立１次方程式

$$Lq = b \tag{1.14a}$$

$$Ux = q \tag{1.14b}$$

に帰着させて解くものである．ただし 上下の三角行列は次のようなものとする．

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & \cdots & l_{nn} \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & 1 & u_{23} & \cdots & u_{2n} \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & u_{n-1,n} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \tag{1.15}$$

Crout 法の計算手順は，行の入替の不要な簡単な連立 1 次方程式

$$\begin{aligned} 2x_1 - 2x_2 - x_3 &= -2 \\ 3x_1 - 2x_2 + x_3 &= -5 \\ x_1 - 2x_3 &= 5 \end{aligned}$$

表 1.4: Crout 法の計算手順

k	L			U			A ($A \to L+U-I$)			B ($b \to q$)	$I \to L^{-1}$		
		0	0	1			2	−2	−1	−2	1	0	0
0			0	0	1		3	−2	1	−5	0	1	0
				0	0	1	1	0	−2	5	0	0	1
	2	0	0	1	−1	−1/2	2	−1	−1/2	−1	1/2	0	0
1	3		0	0	1		3	−2	1	−5	0	1	0
	1			0	0	1	1	0	−2	5	0	0	1
	2	0	0	1	−1	−1/2	2	−1	−1/2	−1	1/2	0	0
2	3	1	0	0	1	5/2	3	1	5/2	−2	−3/2	1	0
	1	1		0	0	1	1	1	−2	5	0	0	1
	2	0	0	1	−1	−1/2	2	−1	−1/2	−1	1/2	0	0
3	3	1	0	0	1	5/2	3	1	5/2	−2	−3/2	1	0
	1	1	−4	0	0	1	1	1	−4	−2	−1/4	1/4	−1/4

B ($\to x$)	$\to A^{-1}$		
1	−1/2	1/2	1/2
3	−7/8	3/8	5/8
−2	−1/4	1/4	−1/4

に対しては表 1.4 のようになる．まず LU 分解から見ていこう．式 (1.13) の成分の式は

$$
\begin{aligned}
&a_{11} = l_{11} \quad a_{12} = l_{11}u_{12} && a_{13} = l_{11}u_{13} \\
&a_{21} = l_{21} \quad a_{22} = l_{21}u_{12} + l_{22} && a_{23} = l_{21}u_{13} + l_{22}u_{23} \\
&a_{31} = l_{31} \quad a_{32} = l_{31}u_{12} + l_{32} && a_{33} = l_{31}u_{13} + l_{32}u_{23} + l_{33}
\end{aligned}
$$

のようになる．$k = 0$ の欄には左から順に L, U の既知の部分，A，b，I を示す．$k = 1$ 段階では，L の第 1 列 l_{l1}, l_{21}, l_{31} と U の第 1 行 u_{12}, u_{13} の値が上式から求められる．$k = 2$ 段階では，L の第 2 列 l_{22}, l_{32} と U の第 2 行 u_{23} の値が求められる．また $k = 3$ 段階では L の第 3 列 l_{33} の値が求められる．この表の 4 列めの欄は，三角行列 L, U の 0 と 1 の記載を省略し，行列 A の用済みの要素を新たに計算した L と U の要素に逐一置き換えたもので，これより LU 分解は 1 つの配列 A 上で実行可能なことが分かる．

次に式 (1.14) による三角行列の連立 1 次方程式の解き方について見ていこう．式 $A = LU$，$b = Lq$，$I = LL^{-1}$ の類似性に注意すれば，q と L^{-1} の値は，U を A から計算したのと全く同様の操作で，b または I から計算することができる．第 5 列めの欄は q，第 6 列めの欄は L^{-1} の計算過程を示す．また $k = 3$ の欄は左から順に L，U，$L+U-I$，q，L^{-1} を示す．表の右下の部分は，第 1 列めが式 (1.14b) による x の計算で，計算は下から順に行われる．第 2 列めは逆行列 A^{-1} の計算で，式 $q = Ux$，$L^{-1} = UA^{-1}$ の類似性から，逆行列 A^{-1} は，x を q から計算したのと全く同様の操作で L^{-1} から計算することができる．

Crout 法の計算は表 1.4 の計算手順を一般化したもので，LU 分解の計算は，式 (1.13) から導かれる式

$$
l_{ik} = a_{ik} - \sum_{\nu=1}^{k-1} l_{i\nu}u_{\nu k} \qquad (k = 1, 2, \ldots, n;\ i = k, \ldots, n) \tag{1.16a}
$$

$$
u_{kj} = \frac{1}{l_{kk}}\Big(a_{kj} - \sum_{\nu=1}^{k-1} l_{k\nu}u_{\nu j}\Big) \qquad (k = 1, 2, \ldots, n-1;\ j = k+1, \ldots, n) \tag{1.16b}
$$

を用い，l_{i1} $(i = 1, \ldots, n)$, u_{1j} $(j = 2, \ldots, n)$, $\ldots$, l_{ik} $(i = k, \ldots, n)$, u_{kj} $(j = k+1, \ldots, n)$, $\ldots$, l_{nn} の順に行われる．LU 分解はすべての主対角要素の大きさ $|l_{kk}|$ がある小さい正数 ε よりも大きいときに可

能である．また 2 つの三角行列の連立 1 次方程式は，式 (1.14) から導かれる次式によって解かれる．

$$q_k = \frac{1}{l_{kk}}\Big(b_k - \sum_{\nu=1}^{k-1} l_{k\nu} q_\nu\Big) \qquad (k = 1, 2, \ldots, n) \tag{1.17a}$$

$$x_k = q_k - \sum_{\nu=k+1}^{n} u_{k\nu} x_\nu \qquad (k = n, \ldots, 2, 1) \tag{1.17b}$$

Crout 法のプログラムは次のようになる．

```
!*********************************************************
!  Solution of system of linear equations by Crout method
!*********************************************************
SUBROUTINE CROUTO(a,n)                                    !LU-decomposition
DIMENSION  a(n,n)
cycle_1: DO k=1,n; IF(k==1)             GOTO 11
  DO i=k,n
    DO nu=1,k-1
      a(i,k)=a(i,k)-a(i,nu)*a(nu,k)                       !compt (1.16a)
    ENDDO
  ENDDO
  IF(k==n)CYCLE cycle_1
  11 DO j=k+1,n; IF(k==1)               GOTO 12
    DO nu=1,k-1
      a(k,j)=a(k,j)-a(k,nu)*a(nu,j)                       !compt (1.16b)
    ENDDO; 12 a(k,j)=a(k,j)/a(k,k)                        !compt (1.16b)
  ENDDO
ENDDO cycle_1
END
!
SUBROUTINE CROUT(a,b,n)                                   !Compt triangular matrix eqs
DIMENSION  a(n,n),b(n)
DO k=1,n; IF(k==1)                      GOTO 21
  DO nu=1,k-1
    b(k)=b(k)-a(k,nu)*b(nu)                               !compt (1.17a)
  ENDDO; 21 b(k)=b(k)/a(k,k)                              !compt (1.17a)
ENDDO
DO k=n-1,1,-1
  DO nu=k+1,n
    b(k)=b(k)-a(k,nu)*b(nu)                               !compt (1.17b)
  ENDDO
ENDDO
END
```

CROUTO は行列 A の LU 分解のサブルーチンで，配列 a に A の値を入れて計算を実行すれば，a に $L+U-I$ の値が出力される．また CROUT は 2 つの三角行列の連立 1 次方程式を解くサブルーチンで，a に $L+U-I$，配列 b に b の値を入れて計算を実行すれば，a は $L+U-I$ のまま，b に x の値が出力される．このプログラムの説明は省略する．なおこのプログラムは行の入替の必要な場合には使えない．

一般に帯行列 A の逆行列 A^{-1} は帯行列にはならないが，$L+U-I$ は式 (1.16) からも分かるように A と同じバンド幅の帯行列になる．したがって，Crout 法は，帯行列の同じ係数行列を持ち定数ベクトルの異なるいくつかの連立 1 次方程式を解く目的に適っている．正方行列 A の Crout 法の計算式 (1.16a)，(1.16b)，(1.17a)，(1.17b) は，縮小行列 $\tilde{A}$ に対しては $a_{ij} = \tilde{a}_{i,j-i}$ また $l_{ij} = \tilde{l}_{i,j-i}$，$u_{ij} = \tilde{u}_{i,j-i}$ であるから次のよ

うに書換えられる．LU 分解の式は

$$\tilde{l}_{i\lambda} = \tilde{a}_{i\lambda} - \sum_{\nu=\nu_1}^{k-1} \tilde{l}_{i,\nu-i}\tilde{u}_{\nu,k-\nu} \qquad (k = 1, 2, \ldots, n;\ i = k, \ldots, i_2) \tag{1.18a}$$

$$\tilde{u}_{k\mu} = \frac{1}{\tilde{a}_{k0}}\Big(\tilde{a}_{k\mu} - \sum_{\nu=\nu_2}^{k-1} \tilde{l}_{k,\nu-k}\tilde{u}_{\nu,j-\nu}\Big) \qquad (k = 1, 2, \ldots, n-1;\ j = k+1, \ldots, j_2) \tag{1.18b}$$

ただし $\lambda = k-i,\ \mu = j-k$ である．第 k 段階で，式 (1.18a) で $\tilde{l}_{i\lambda}$ を計算する範囲は，図 1.1 に示すような帯行列 A と縮小行列 $\tilde{A}$ 上で検討すれば，$k \le i \le n$ かつ $m_1 \le k-i$ であるから $i_2 = \min(n,\ k-m_1)$ になる．また ν に関して総和を取る範囲は，$1 \le \nu \le k-1$ であるとともに $\tilde{l}_{i,\nu-i}$ の $\nu-i$ が $m_1 \le \nu-i$，$\tilde{u}_{\nu,k-\nu}$ の $k-\nu$ が $m_2 \ge k-\nu$ でなければならないので，$\nu_1 = \max(1,\ i+m_1,\ k-m_2)$ となる．同様に式 (1.18b) で $\tilde{u}_{k\mu}$ を計算する範囲は $k+1 \le j \le n$ かつ $j-k \le m_2$ であるから $j_2 = \min(n,\ k+m_2)$，総和を取る範囲は $1 \le \nu \le k-1$，$m_1 \le \nu-k$，$m_2 \ge j-\nu$ であるから $\nu_2 = \max(1,\ k+m_1,\ j-m_2)$ となる．

$$q_k = \frac{1}{\tilde{l}_{k0}}\Big(b_k - \sum_{\nu=\nu_2}^{k-1} \tilde{l}_{k,\nu-k}q_\nu\Big) \qquad (k = 1, 2, \ldots, n) \tag{1.19a}$$

$$x_k = q_k - \sum_{\nu=k+1}^{j_2} \tilde{u}_{k,\nu-k}x_\nu \qquad (k = n, \ldots, 2, 1) \tag{1.19b}$$

ただし $\nu_2 = \max(1,\ k+m_1),\ j_2 = \min(n,\ k+m_2)$ である．式 (1.19) の計算で総和を取る範囲は $\tilde{a}_{k,\nu-k}$ の $\nu-k$ が $m_1 \le \nu-k \le m_2$ の範囲にあるように制限される．

縮小行列 $\tilde{A}$ に対する Crout 法のプログラムは次のようになる．

```
!***********************************************************
!  Solution of system of linear equations with band matrix
!  by Crout method using reduced array
!***********************************************************
SUBROUTINE CROUTBO(at,n,m1,m2)              !LU-decomposition
DIMENSION  at(n,m1:m2)
cycle_1: DO k=1,n; IF(k==1)             GOTO 11
  i2=MIN0(n,k-m1)
  DO i=k,i2; la=k-i; nu1=MAX0(1,i+m1,k-m2)
    DO nu=nu1,k-1
      at(i,la)=at(i,la)-at(i,nu-i)*at(nu,k-nu)              !compt (1.18a)
    ENDDO
  ENDDO
  IF(k==n)CYCLE cycle_1
  11 j2=MIN0(n,k+m2)
  DO j=k+1,j2; mu=j-k; nu2=MAX0(1,k+m1,j-m2)
    DO nu=nu2,k-1
      at(k,mu)=at(k,mu)-at(k,nu-k)*at(nu,j-nu)              !compt (1.18b)
    ENDDO; at(k,mu)=at(k,mu)/at(k,0)                        !compt (1.18b)
  ENDDO
ENDDO cycle_1
END
!
SUBROUTINE CROUTB(at,b,n,m1,m2)                             !Compt triangular matrix eqs
DIMENSION  at(n,m1:m2),b(n)
DO k=1,n; IF(k==1)                      GOTO 21
  nu2=MAX0(1,k+m1)
  DO nu=nu2,k-1
```

```
    b(k)=b(k)-at(k,nu-k)*b(nu)                          !compt (1.19a)
  ENDDO; 21 b(k)=b(k)/at(k,0)                           !compt (1.19a)
ENDDO
DO k=n-1,1,-1; j2=MIN0(n,k+m2)
  DO nu=k+1,j2
    b(k)=b(k)-at(k,nu-k)*b(nu)                          !compt (1.19b)
  ENDDO
ENDDO
END
```

CROUTB0は行列 $\tilde{A}$ を LU 分解するサブルーチンで，配列 at に $\tilde{A}$ の値を入れて計算を実行すれば at は L+U−I になる．また CROUTB は 2 つの三角行列の連立 1 次方程式を解くサブルーチンで，配列 at に L+U−I，配列 b に b の値を入れて計算を実行すれば，at は L+U−I のままで，b に解 x の値が出力される．このプログラムでは配列 at の主対角要素の列番号を 0，左端と右端の列番号を m1,m2 としている．なおサブルーチン CROUTB0 と CROUTB は正方行列のサブルーチン CROUT0 または CROUT に対応するように作られている．

ここで各種の直接法の計算時間を比較する．表 1.5 は Gauss-Jordan 法，Gauss 消去法，Crout 法とともに Cramer 法の実数の四則演算回数を示したものである．代数の教科書によく出てくる Cramer 公式は次式で与えられる．

$$x_i = \frac{1}{|A|}\sum_{j=1}^{n} A_{ji}b_j \qquad (i = 1,\, 2,\, \ldots,\, n) \tag{1.20}$$

ただし A_{ji} は行列 A の要素 a_{ji} の余因子である．Cramer 法の演算回数は $(n+2)!$ と言われ膨大である．いま小行列式に分解して計算することにすれば，実数の演算回数は筆者の見積もりでは表に示すようにこれよりは少なくなる．他方 行列式を消去法で計算すれば更に少なくなるが，それでも他の方法に較べて多く Cramer 法が実際の計算に用いられることはない．Gauss 消去法と Crout 法は係数行列が帯行列の場合に適し，バンド幅の中だけを縮小行列に入力し必要なメモリを大幅に節約することができる．また同じ係数行列を持ち定数ベクトルの異なるいくつかの方程式を解く場合には， 逆行列を用いる方法または Crout 法が適している．Crout 法は計算手順が複雑で演算回数も多いように思われるが，実際には最も演算回数の少ない方法である．しかもその多くは LU 分解に費やされるので，帯行列の同じ係数行列を持ち定数ベクトルの異なるいくつかの方程式を解く場合にはたいへん重宝である．

表 1.5: Cramer 法，Gauss-Jordan 法，Gauss 消去法，Crout 法の実数演算回数

解　　法	実数演算回数	$n=5$	$n=10$	$n=100$
Cramer 法 (小行列式)	$2.7(n+1)!$	1,949	109×10^6	
Cramer 法 (消去法)	$2n^4/3$	425	6,775	
Gauss-Jordan 法	n^3	135	1,045	$1,015\times10^3$
Gauss 消去法	$2n^3/3$	115	805	682×10^3
Crout 法	$2n^3/3$		769	677×10^3
(CROUT0)	$2n^3/3$		579	657×10^3
(CROUT)	$2n^2$		190	20×10^3

最後に，ここに載せたいくつかのプログラムはすべて FORTRAN 90/95 で compile し error や warning のないこを確認し，更に簡単な例題を解き正しい結果の得られることも確かめている．しかしながらこれらのプログラムに誤りのないことを保障するものではない．