Bull. Mukogawa Women's Univ. Nat. Sci., **39**, 1−7(1991)
武庫川女子大紀要（自然科学）

数値計算における定義による算法の実用性について
—— Cramerの法則を用いた連立一次方程式の数値解法 ——

丸 山 健 夫

（武庫川女子大学文学部教育学科人間関係コース）

Practicability of Numerical Solution according to the Original Definitions
——Numerical Solution of Simultaneous Linear Equations using Cramer's Rule——

Takeo Maruyama

Department of Human Relations, Faculty of Letters, Mukogawa Women's University, Nishinomiya 663, Japan

Using Cramer's rule in solving n simultaneous equations in n unknowns, $n+1$ determinants should be evaluated. For this reason Cramer's rule is said to be of no practical use. But this solution is the most fundamental way and the original approach to solve a set of simultaneous linear equations.

The practicability of numerical solution using Cramer's rule was discussed in this paper. It was revealed that this method can be put to practical use by personal computers and has many merits compared with other numerical methods. The data for the examination of the performance of computers were also obtained.

1．緒　言

連立一次方程式の解法には，Cramer の法則による古典的解法のほか，Gauss 法，Gauss－Jordan 法などの消去法や Jacobi 法，Gauss－Seidel 法に代表される反復法による数値解法がある．

このうち Cramer の法則による解法（以下 Cramer 法）は，最も理論的で一般性のある解法である．しかし，n 元の場合で n 次行列式を $(n+1)$ 個計算する必要があることから，たいていの文献では 3 元あるいは 4 元以上では実用的でないとの記述が一般的である．[1),2)]

今日，パーソナルコンピュータの普及により，時間的にそして経済的にも CPU を個人が占有できる環境にある．そこで，最も平均的に使用されている機種とプログラム言語を使用して，Cramer 法の実用性についての検討を行った．その結果，7 元程度までであればかなりの時間的な実用性があることがわかった．しかも，ピボット選択や収束条件を考慮する必要がなく確実に解ける上に，計算精度もきわめて高いことがわかった．

また，これらの比較的計算量が多く，他のハードウェアやソフトウェアでも同じアルゴリズムで実行可能な計算をさまざまな実行環境で検討したことにより，ハードウェアおよびソフトウェアの性能試験の参考資料となるデータも得られた．

2．方　法

2.1　Cramer 法

Cramer 法はつぎのような公式を基礎としている．

n 元連立一次方程式の係数行列を $A=(a_{ij})$ 定数項を (b_j) とすると，解 x_j は，

$$x_j=\frac{|\ A_j\ |}{|\ A\ |}=\frac{\begin{vmatrix} a_{11}\cdot b_1\cdot a_{1n} \\ a_{21}\cdot b_2\cdot a_{2n} \\ \vdots \quad\quad \vdots \\ a_{n1}\cdot b_n\cdot a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11}\cdot a_{1j}\cdot a_{1n} \\ a_{21}\cdot a_{2j}\cdot a_{2n} \\ \vdots \quad\quad \vdots \\ a_{n1}\cdot a_{nj}\cdot a_{nn} \end{vmatrix}}$$

と書ける.[3] ここで A_j は係数行列式 A の第 j 列を定数項 (b_j) で置き換えた行列である．

また行列式については，$\{1\quad 2\quad \cdots\quad n\}$ という n 文字の置換を $\sigma=\{\sigma_1\sigma_2\cdots\sigma_n\}$ とし，$\varepsilon(\sigma)$ を σ が偶置換のとき $+1$，σ が奇置換のとき -1 をとる係数とするとき，A を例にとればつぎのように定義される.[4] ただし，Σは全置換σにわたる和を表す．

$$\begin{vmatrix} a_{11}\ a_{12}\cdot\cdot a_{1n} \\ a_{21}\ a_{22}\cdot\cdot a_{2n} \\ \vdots \quad\ \vdots \quad\quad \vdots \\ a_{n1}\ a_{n2}\cdot\cdot a_{nn} \end{vmatrix}=\Sigma\varepsilon(\sigma)a_{1\sigma_1}a_{2\sigma_2}\cdots a_{n\sigma_n}$$

以上のアルゴリズムを忠実に実行するプログラムを作成して，その実用性を検討した．

ハードウェアは武庫川女子大学人間関係学科で教育用機器として採用されている機種であるパーソナルコンピュータ PC−9801VX を使用した．

プログラム言語は，最も一般的なプログラム言語である N−88 BASIC(86) を用いた．

2.2　Ⅰ型プログラム

Cramer 法において，最も計算量の多い部分はすべての置換を算出するルーチンである．最も単純な計算手順をとればつぎのようになる．n 元の場合なら重複を許した n 個の要素からなる順列を n^n 個発生させて，その中から互いに重複のない順列である置換を取り出し，その置換を用いて定義におけるΣの各項の値を計算して和を累積していく．

5 元の場合の BASIC プログラムを示すと Fig.1 のようになる．

この最も原始的な考え方によるプログラムを 2 元から 7 元まで 6 種類作成し，Ⅰ型とした．

2.3　Ⅱ型プログラム

Ⅰ型のプログラムでは，n 元の場合ならすべての置換を導き出すために n 重ループからなる入れ子構造ができる．置換の総数は，$n!$ であるのに対して，この入れ子による計算ルーチンは，n^n 回も実行される．そこで，入れ子の中に深く入らないうちに前もって重複順列をチェックして無駄な計算を省略するという改良を行った．具体的には，Fig.2 のようなルーチンを各ループの直後に挿入した．このタイプのプログラムを 2 元から 7 元まで 6 個作成し，Ⅱ型とした．

2.4　Ⅲ型プログラム

Ⅰ型，Ⅱ型ともに，置換の算出はそれぞれの実行ごとに行っている．つまり，n 元の場合 $(n+1)$ 回の行列式の計算ごとに $n!$ 個の置換を導き出す同じ計算をしている．

そこで，各元の場合の置換は一定であるのだから，このデータテーブルをあらかじめファイルとしてつくっておき，個々の行列式の計算ではこのファイルから置換に関するデータを読み込み活用する形式で実行速度を高める方法が考えられる．

ここで保存する置換に関するデータファイルは，1 つの置換を表す順列とその偶奇性を判断した $+1$ または -1 という値により 1 つのレコードが構成される．

そこで，この方法で計算を実行するには，データファイルを作成するプログラムと作成されたデータファイルから置換データを読みだし行列式の計算を実行するプログラムの 2 種類のプログラムが必要となる．

この方法をⅢ型とし，前者に添字 M を後者に R をつけて，それぞれのプログラムをⅢ型 M，Ⅲ型 R として区別する．

また，他の型のプログラムについても，たとえばⅠ型で 5 元連立一次方程式を計算するプログラムの場合，Ⅰ型 (5) のプログラムなどと略記する．

実際の計算は，インタプリタとコンパイラの 2 通りで実行した．そこでそれぞれの実行を区別するため，プログラムⅠ型 (5) をインタプリタで実行した場合をⅠ型 (5)（インタプリタ），コンパイラで実行した場合には，Ⅰ型 (5)（コンパイラ）などと記すことにする．

以上により，プログラムはⅠ型からⅢ型 R までそれぞれ 2 元から 7 元まで作成したため，合計 24 個となる．また，実行はそれぞれについてインタプリタおよびコンパイラを使用した 2 種類のため，合計 48 種類の実行結果が得られる．

```
10    CRAMEL METHOD
20   DIM A(5,6),B(5,5),C(5)
30   DIM S(5),SS(5),M(5,6)
40   FOR I=1 TO 5
50     FOR J=1 TO 6
60     PRINT STR$(I)+" 行"+STR$(J)+"列は";:INPUT X
70       A(I,J)=X
80     NEXT
90   NEXT:PRINT
100   FOR I=1 TO 5:FOR J=1 TO 6
110     PRINT A(I,J),
120  NEXT:PRINT:NEXT:PRINT
130   FOR II=1 TO 5
140    FOR JJ=1 TO 5
150     M(II,JJ)=A(II,JJ)
160    NEXT
170   NEXT
180  GOSUB 1000
190   D5=D
200  PRINT "D5=";D5:PRINT
210  FOR P=1 TO 5
220    FOR I=1 TO 5
230      FOR J=1 TO 5
240        B(I,J)=A(I,J)
250      NEXT
260    NEXT
270  FOR II=1 TO 5
280   B(II,P)=A(II,6)
290  NEXT
300   FOR II=1 TO 5
310    FOR JJ=1 TO 5
320     M(II,JJ)=B(II,JJ)
330    NEXT
340   NEXT
350  GOSUB 1000
360   XX=D
370  X=XX/D5
380   PRINT "X=";X
390  NEXT
400  END
1000   'DET5
1010   D=0
1020   FOR I1=1 TO 5
1030    FOR I2=1 TO 5
1040     FOR I3=1 TO 5
1050      FOR I4=1 TO 5
1060       FOR I5=1 TO 5
1070        SS(1)=0:SS(2)=0:SS(3)=0:SS(4)=0:SS(5)=0
1080        S(1)=I1:SS(I1)=1
1090        S(2)=I2:SS(I2)=1
1100        S(3)=I3:SS(I3)=1
1110        S(4)=I4:SS(I4)=1
1120        S(5)=I5:SS(I5)=1
1130        CH=SS(1)*SS(2)*SS(3)*SS(4)*SS(5)
1140       IF CH=0 THEN 1240
1150       C=0
1160        FOR I=1 TO 5
1170         II=I
1180         IF S(II)<>I THEN II=II+1:GOTO 1180
1190         IF II=I THEN 1200 ELSE W=S(I):S(I)=S(II):S(II)=W:C=C+1:GOTO 1200
1200        NEXT
1210       IF C/2=C¥2 THEN DD=1 ELSE DD=-1
1220       DD=DD*M(1,I1)*M(2,I2)*M(3,I3)*M(4,I4)*M(5,I5)
1230        D=D+DD
1240       NEXT
1250      NEXT
1260     NEXT
1270    NEXT
1280   NEXT
1290   RETURN
```

Fig. 1. Example of BASIC Program using Cramer's Rule (5 unknowns)

```
1020   FOR I1=1 TO 5
1030    FOR I2=1 TO 5
1032      U(1)=0:U(2)=0:U(3)=0:U(4)=0:U(5)=0
1034      U(I1)=1:U(I2)=1
1036      R=U(1)+U(2)+U(3)+U(4)+U(5)
1038      IF R<2 THEN 1270
1040     FOR I3=1 TO 5
1042      U(1)=0:U(2)=0:U(3)=0:U(4)=0:U(5)=0
1044      U(I1)=1:U(I2)=1:U(I3)=1
1046      R=U(1)+U(2)+U(3)+U(4)+U(5)
1048      IF R<3 THEN 1260
            ・・・・・・・・・・・・

1240       NEXT
1250      NEXT
1260     NEXT
1270    NEXT
1280   NEXT
```

Fig. 2. Improvement of the BASIC Program

3．結　果

計算を実行した環境についてまとめるとTable 1のようになる。各プログラムを実行して、Table 2を得た。

Table 1. Conditions of Experiments

Model	NEC PC−9801VX
C.P.U.	80286(i80286)
Clock	10 MHZ
Operating System	MS−DOS(Version 3.10)
Interpreter	NEC N−88 BASIC(86) (version 5.0, MS−DOS version)
Compiler	Microsoft BASIC Compiler (Version 5.35)

なおコンパイルおよびリンクのための時間は省略し，実行時のみの時間を測定した．

計算実行時間の測定は，係数データのキーボードからの入力が終了した時点を起点とし，結果のディスプレイ画面への表示が終了した時点までのあいだの時間を測定した．

具体的には，Fig.1のⅠ型(5)の場合を例にとれば，95行目に

95 TIME$＝“00:00:00”

を，395行目に

395 PRINT TIME$

を挿入して行った．

また，実行はPC−9801VXを2台使用して，並行して同じ計算をして，計算結果や実行時間に相違がないか確認しながら行った．実行時間については，1秒程度のずれが生じる場合もまれに起こった．その場合は大きいほうの数値を採用してデータに＊印を付けた．

また，Ⅰ型(7)(インタプリタ)については，23時間28分35秒と23時間22分23秒という2つの結果を得たので算術平均値を結果として示した．

なお，実行時間が1秒未満の場合，−印で示した．

4．考　察

4.1　実行環境の考察

実行環境の違いによる計算時間の相違について検討するため，Fig.1にあるⅠ型(5)のプログラムを基準としてさまざまな環境のもとでインタプリタによる実行を行った．

Table 2. Time taken to Solve a Set of Simultaneous Linear Equations

TYPE Ⅰ						
Oder	2	3	4	5	6	7
Interpreter	−	1s	13s	3m22s	1h03m08s	23h25m29s
Compiler	−	−	1s	16s	4m50s	1h46m13s
TYPE Ⅱ						
Oder	2	3	4	5	6	7
Interpreter	−	1s	8s	1m05s	10m08s	1h44m18s
Compiler	−	−	1s	6s	1m00s	10m29s
TYPE Ⅲ M						
Oder	2	3	4	5	6	7
Interpreter	−	−	2s	12s	1m31s	13m39s*
Compiler	−	−	−	2s	12s	1m49s
TYPE Ⅲ R						
Oder	2	3	4	5	6	7
Interpreter	−	−	1s	11s	1m03s	9m09s
Compiler	−	−	1s	8s	28s	3m29s

まず，N－88 BASIC(86) はそのまま同一にしておきOSであるMS－DOSについて，バージョンを3.10のほか2.11,3.3Cと変えて実行した．しかし，バージョンの違いによる計算時間の相違は認められなかった．

つづいて，CPUとクロック周波数の違いについて検討した.[5] PC－9801VXは，2つのCPUを搭載し，それぞれクロック周波数も2種類が選択できるため，合計4種類の条件での実行が可能である．Ⅰ型(5)の実行結果は，Table 3のようであった．

Table 3. Relationship between CPU, Clock Frequency and Time for Solution

CPU	Clock	Time
80286 (i80286)	10MHZ	3m22s
	8MHZ	4m11s
V30 (μPD70116－10)	10MHZ	6m43s
	8MHZ	8m29s

BASICと呼ばれる言語は，文法はほぼ共通であるが実行や編集をおこなうインタプリタやコンパイラそしてエディタを含めたプログラム群には，いくつかの異なる製品が市販されている．これらのBASICの違いによる実行速度の相違について，やはり同じⅠ型(5)のプログラムを用いて測定して，Table 4の結果を得た．プログラムはすべてFig.1のソースプログラムを使用した．

Table 4. Difference between The Six BASIC Languages

N－88 BASIC(MS－DOS)	3m22s
N－88 BASIC(compiler)	16s
N－88 BASIC(DISK)	3m12s
Q－BASIC(interpreter)	1m03s
Q－BASIC(compiler)	27s
U－BASIC(interpreter)	56s*

使用したBASICはつぎの6種類である．

(1) N－88 BASIC(MS－DOS)
NEC N－88 BASIC(86) MS－DOS version (Version 5.0)

(2) N－88 BASIC(compiler)
Microsoft BASIC Compiler (Version 5.35)

以上の(1)(2)の2つは，すべての型での実行に使用して，Table 2を得たインタプリタとコンパイラである．

(3) N88－BASIC(DISK)
NEC N－88 BASIC(86) Disk version (Version 3.0)

(4) Q－BASIC(interpreter)
Microsoft QuickBASIC (version 4.2)

(5) Q－BASIC(compiler)
(4)のQuickBASICに添付されているQuickBASIC KANJI Compiler Version 4.20を利用してランタイム分離型EXEファイルを作成

(6) U－BASIC(interpreter)
木田[6]の作成による多倍長計算用BASIC (Version 8.11)

Ⅲ型についてはフロッピーディスクに対するデータの読み書きがおこなわれる．そこで最も大量のデータをやりとりするⅢ型(7)についてバッファ領域の大きさの違いによる計算時間の比較を行ってTable 5を得た．結果としては，あまり相違がないので既定値である2(BUFFERS＝2)をTable 2の結果では用いている．

Table 5. Relationship between Buffers and Time for Solution

Buffers	Type Ⅲ M(7)	Type Ⅲ R(7)
2	13m39s*	9m09s
5	13m39s*	9m06s*
10	13m39s*	9m05s*
20	13m39s*	9m05s*
30	13m39s	9m05s*
50	13m39s	9m05s*

4.2 型別比較

Table 1の結果について，型別の考察を行う．

Ⅰ型は，最も計算量の多い型である．Ⅰ型(4)(インタプリタ)では13秒，Ⅰ型(5)(インタプリタ)で3分22秒の計算時間がかかっている．Ⅰ型(6)(コンパイラ)で4分50秒であり，Ⅰ型ではインタプリタで5元，コンパイラ使用で6元が実用上の限界であろう．

Ⅱ型では，Ⅱ型(5)(インタプリタ)ではⅠ型の3分の1以下の1分05秒に，Ⅱ型(7)(インタプリタ)もⅠ型(7)(インタプリタ)の13分1以下の1時間44分18秒に短縮された．

Ⅱ型(5)(インタプリタ)とⅡ型(6)(コンパイラ)が1分程度で，またⅡ型(6)(インタプリタ)とⅡ型(7)(コンパイラ)が10分程度でほぼ同時間である．Ⅱ型では，コンパイルすると同じ実行時間で1元上の計算が可能となる．

Ⅲ型については，Ⅲ型R(5)(コンパイラ)の8秒は注目に値する．Ⅱ型(5)(コンパイラ)が6秒であることから，Ⅱ型からⅢ型への改良の効果がないことになる．これは，ディスクへのアクセス時間の影響と考えられる．ただし，6元以上では明らかに改良の効果がみられる．

Ⅲ型R(7)(コンパイラ)の計算時間は，3分29秒であった．この程度までであれば実用上支障がないと考えられる．

しかしⅢ型には，データファイルの大きさが元の数が増すごとに膨大な量となるという問題点がある．Ⅲ型ではフロッピーディスクにシーケンシャルファイルとして

PRINT #1, DD; I1; I2; I3; I4; I5

という形式でデータを書き込んだ．この場合，作成される置換データファイルの総バイト数はn元のとき，

(総バイト数)$=(3(n+1)+2)n!+1$

で与えられ，10元までを計算するとTable 6のようになる．

Table 6. Example of Memory Size for Data File

n	$(3(n+1)+2)n!+1$
1	9
2	23
3	85
4	409
5	2401
6	16561
7	131041
8	1169281
9	11612161
10	127008001

ただし，コンパイラ使用の場合は，128バイト単位の処理が行われるので128の倍数に修正される．

この計算からIMB(最大1250304バイト)の容量を持つ一般的な2HDフロッピーディスクを用いた場合，実用上は8元が限界となる．

以後n元で$n!$レコードが必要となり，ハードディスクやマルチボリュームファイルを用いても，必ずいつかは限界がでる．この点がⅢ型の最大の欠点である．

4.3　他の計算手法との比較

Cramer法による計算と，他の数値計算手法との比較検討を行った．消去法の中からGauss法を，反復法からJacobi法を取り上げ，N−88 BASIC(86)によりプログラムを作成し，同一の連立一次方程式について問題解法を実行して比較検討を行った．

まず，計算時間の比較では，Ⅰ型(5)(インタプリタ)および，Gauss法，Jacobi法を用いたN−88 BASIC(86)のインタプリタによる実行の3種類を比較した．

結果はTable 7のようであるが予想通りGauss法やJacobi法によるプログラムのほうがはるかに時間的には優れていることがわかる．しかし，Gauss法ではピボットの選択を行う必要がある．またJacobi法ではピボット選択に加えて，係数行列が収束条件を満足する必要がある．

Table 7. Difference between The Tree Methods

Method	Time
Gauss	—
Jacobi	3s
Type I(5)	3m22s

これに対して，Cramer法は係数行列に対する制限は全くない．連立一次方程式の解が存在するときは必ず解法できる．

つぎに，つぎのような3元連立一次方程式の解法を通して，計算精度の問題を検討した．[7]

$$15.0x_1+15.0x_2+14.0x_3=58.0$$
$$15.0x_1+14.0x_2+13.0x_3=55.0$$
$$14.0x_1+13.0x_2+12.0x_3=51.0$$

Gauss法によるN−88 BASIC(86)プログラムでの計算結果は，

$(x_1,x_2,x_3)=(0.999998,1.00003,1.99997)$

となり，理論的な解（1,1,2）とは異なる答が導かれる．これは，消去法による計算の過程での丸め誤差の累積効果による．しかし，同じ問題をCramer法によるⅠ型（3)（インタプリタ）で解法すると結果は，（1,1,2）となり真の解が導き出される．

この場合の計算時間はGauss法で1秒未満であるがCramer法でもTable 2より1秒台であり実用上大差ない．

また，同じ問題のJacobi法での解法は，収束条件が満たされずこのままでは不可能である．

以上，連立一次方程式のCramer法を定義どおりに用いたコンピュータによる数値解法についてまとめるとつぎのようになる．

(1) 次元ごとにプログラムを別個に作成しなければならない．
(2) 計算時間は7元程度までであれば実用上問題ない．
(3) Ⅲ型では，データファイルの大きさから，8元程度が限界である．
(4) 消去法や反復法にみられるピボット選択や収束するための条件などの係数行列に対する制約を全く考慮する必要がない．
(5) 消去法にみられる丸め誤差の影響もほとんどなく，非常に精度の高い解が得られる．
(6) 係数行列が正則でない場合を除き，すなわち代数的に解が求められない場合以外，解は確実に求められる．

5．文　献

1) マッカーラ，T.R.（三浦功・田尾陽一共訳），計算機のための数値計算法概論，サイエンス社，東京，p.146(1972)
2) 戸川隼人，計算機のための数値計算，サイエンス社，東京，p.17(1976)
3) 斎藤正彦，線型代数入門，東京大学出版会，東京，p.89(1966)
4) 前掲3)，pp.77－78(1966)
5) 日本電気，PC－9801VX ハードウェアマニュアル，NEC, 東京，pp.19－20(1987)
6) 木田祐司，UBASIC86 Ver.8.1 ユーザーズマニュアル，日本評論社，東京，(1990)
7) 前掲1)，p.150(1972)