

多元連立一次方程式の数値解法とプログラム

水 島 章 次*, 坂 田 宜 子**

Computer Program and Numerical Method for Simultaneous Linear Algebraic Equations

Akitsu Mizushima* and Yoshiko Sakata**

(Received September 1, 1969)

We discuss a large scale matrix programming and a convergency of numerical method for simultaneous linear algebraic equations.

Numerical methods used in this work are Gauss Seidel Method, Conjugate Gradient Method, and Steepest Descent Method.

1. ま え が き

電子計算機がわが国に普及しはじめてはや十数年になり、現在ではほとんどの分野においてなんらかの形で利用されている。このように広く一般に使用されるようになると、電子計算機の適用分野は急激に拡大され、ハードウェアは高速化、大型化の方向へ進んできた。

周知のようにメモリは情報の格納場所であり、一般の利用者にとって最も関係の深い部分である。このために現在のように広く利用されるようになると、ますます情報量が増加し、必然的にメモリ不足が生じてきた。

ここで科学技術計算のために電子計算機を使用する場合について見ると、そのほとんどは数学的にモデル化した現象の理論的考察や、現象そのものをモデル化して考察するためと考えられる。この数学的手法について見れば、電子計算機による処理が容易な方法がとくに使用されている。たとえば、マトリックスを扱うような場合が代表的な例であろう。筆者らの知る限りでは、土木、建築、航空、造船などの分野ではマトリックス法による構造解析や、偏微分方程式を差分法で解く場合を多く見かけている。これらの計算は最終的に多元連立一次方程式を解くことであり、現象を正確に解析しようとすればするほど多くのメモリを必要とする。しかし、大学関係研究者が日常使用している電子計算機は、各自の大学のものか、あるいは、全国共同利用大型計算機である。これらの電子計算機の大きさは、たとえば、東大大型計算機セ

ンターの HITAC-5020E では内部記憶容量が65キロ語（32ビット／語）で、処理プログラムの格納を考えれば、とりうるマトリックスの大きさは約 200 元である。また中部工業大学電子計算機センターの FACOM-270/30は内部記憶容量が16キロ語（16ビット／語）で約60元のマトリックスである。この程度の大きさのマトリックスでは到底研究者の満足するほどの考察はできない。

上述したようにいかに大きなマトリックスを処理するかということと、つぎに研究者の迷うのは、数多くある多元連立一次方程式数値解法のうちいずれを用いればよいかということである。この解法の選択をあやまれば、解を不安なものにするだけでなく、全然計算できない場合も生ずる。結局、多元連立一次方程式を解く必要のある研究者は、以前自分の使ったことのある方法か、または、人伝てに聞いた方法を使用することになる。

以上のような問題に対し、筆者らは本文でメモリ不足に対する解決策およびプログラミングについて述べ、今までに開発したプログラムおよび二、三の数値実験について興味を持つ方の参考に供したい。

2. マトリックスの表現

一般に大きなマトリックスとは、その計算手段の進歩とともに変わっている。たとえば、二十年程前には10元以上とされたものが、現在では電子計算機存在を前提として数百元～数千元とする人もある。このような大きなマトリックスを記憶させるには、いかに現存の電子計算機といえどいろいろ工夫しなければならない。

すでに述べたように、マトリックス法による構造計算などのマトリックスは、大きいといってもその内容は零

* 基礎教育系列、電子計算機センター (Department of Fundamental Science and Computer Center)

** 電子計算機センター (Computer Center)

要素の方が非零要素よりはるかに多い。そこで以下に述べるような表現法をとれば十分大きなマトリックスを処理できる。

いま、表現すべきマトリックスを n 元とし、非零要素の個数が最大の行に注目し、その個数を m とする。そこで図-1のようにマトリックスを変換する。この変換によりメモリは n^2 個から $n \times m$ 個になり、 m が小さいほど n は大きくとれる。この実際例について見れば、たと

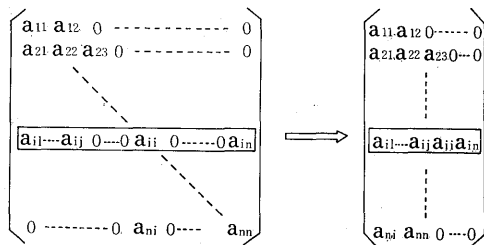


図-1 マトリックスの変換

えば、すでに述べたマトリックス法による構造計算のうち、最近、二次元問題などの解析法として有限要素法が多く使用されているが、最も実用的に使用されている方法は、1行に対する非零要素の個数はせいぜい20~30個であり、これから考えても十分大きなマトリックスを扱える。

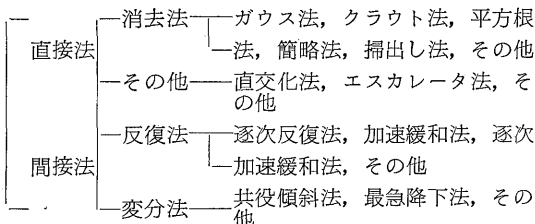
以上述べた方法により、筆者らの開発したプログラムおよび使用した数値解法について次に示す。

3. 数値解法およびプログラム

多元連立一次方程式の数値解法は実に多くの種類があり、その解法の選択は大変なものである。

1) 解法の分類および選択

数値解法は大別して、直接法、間接法の二種類に分けられる。



現在、これらの解法を使用する場合、使い分ける適確な指針はない。直接法にすべきか間接法にすべきかが問題となる。しかし次の点は明確である。一般に直接法はほとんどの場合何んらかの解を得るが、間接法はその選択をあやまれば解を得られないことがある。しかし、間接法は、大きなマトリックスの場合には有利である。そこでわれわれは間接法の代表的な解法のプログラムを作

成し、二、三の実験を行なった。作成したプログラムは、逐次反復法、共役傾斜法、最急降下法に関するもので、2.で述べたように零要素をとり出したマトリックスを処理するものである。以下説明を簡単にするために式中の記号は次のものを使用する。

A	係数行列
A^T	A の転置行列
B	定数ベクトル
X	解ベクトル
$AX=B$	問題の方程式
R, P, V	ベクトル
a_{ij}, b_{ij}, x_{ij}	それぞれ A, B, X の要素
α, β	スカラー
n	元数
(ν)	第 ν 近似

2) 逐次反復法およびプログラム

第0近似 $X^{(0)}=B$

$$\text{第}\nu+1\text{近似 } x_i^{(\nu+1)} = (b_i - \sum_{k=1}^{i-1} a_{ik} \cdot x_k^{(\nu+1)} - \sum_{j=i+1}^n a_{ij} \cdot x_j^{(\nu)}) / a_{ii}$$

```

SUBROUTINE GAUSS (A,IR,B,X,XX,N,JN,ICN,
EPS)
DIMENSION A (N,JN),IR(N,JN),B(N),X(N),
XX(N)
IC=0
DO 1000 I=1,N
1000 X(I)=B(I)
1111 IC=IC+1
DO 1110 I=1,N
XX(I)=X(I)
WS=0.0
DO 1100 K=1,JN
J=IR(I,K)
IF(J.EQ.0) GO TO 1110
IF(J.EQ.I) GO TO 1100
WS=WS+A(I,K)*X(J)
1100 CONTINUE
1110 X(I)=B(I)-WS
DO 1210 I=1, N
XI=X(I)
XXI=XX(I)
IF(XI.EQ.0.0) GO TO 1200
IF(ABS(XXI-XI-1.0).GE.EPS) GO TO 1300
1200 IF(ABS(XI-XXI).GE.EPS) GO TO 1300
1210 CONTINUE
RETURN
1300 IF(IC.LT.ICN) GO TO 1111
RETURN
END
  
```

3) 共役傾斜法およびプログラム

第 0 近似 $X^{(0)} = B$
 $R^{(0)} = B - AX^{(0)}$
 $P^{(0)} = R^{(0)}$
 第 $\nu+1$ 近似 $\alpha^{(\nu)} = (P^{(\nu)}, R^{(\nu)}) / (AP^{(\nu)}, P^{(\nu)})$
 $X^{(\nu+1)} = X^{(\nu)} + \alpha^{(\nu)} P^{(\nu)}$
 $R^{(\nu+1)} = R^{(\nu)} - \alpha^{(\nu)} AP^{(\nu)}$
 $\beta^{(\nu)} = (AP^{(\nu)}, R^{(\nu+1)}) / (AP^{(\nu)}, P^{(\nu)})$
 $P^{(\nu+1)} = R^{(\nu+1)} - \beta^{(\nu)} P^{(\nu)}$

SUBROUTINE CONJU(A,IR,B,X,XX,R,P,V,N,
 JN,ICN,EPS)

DIMENSION A(N,JN),IR(N,JN),B(N),X(N),
 XX(N),R(N),P(N),V(N)

IC=0

DO 1000 I=1,N

1000 X(I)=B(I)

DO 1110 I=1,N

WS=0.0

DO 1100 K=1,JN

J=IR(I,K)

IF(J.EQ.0) GO TO 1110

1100 WS=WS+A(I,K)*X(J)

1110 V(I)=WS

DO 1120 I=1,N

R(I)=B(I)-V(I)

1120 P(I)=R(I)

1111 IC=IC+1

BSI=0.0

DO 1200 I=1,N

XX(I)=X(I)

1200 BSI=BSI+P(I)*R(I)

DO 1310 I=1,N

WS=0.0

DO 1300 K=1,JN

J=IR(I,K)

IF (J.EQ.0) GO TO 1310

1300 WS=WS+A(I,K)*P(J)

1310 V(I)=WS

BBO=0.0

DO 1400 I=1,N

1400 BBO=V(I)*P(I)+BBO

ALPHA=BSI/BBO

DO 1410 I=1,N

1410 X(I)=XX(I)+ALPHA*P(I)

DO 1510 I=1,N

XI=X(I)

XXI=XX(I)

IF(XI.EQ.0.0) GO TO 1500

IF(ABS(XXI/XI-1.0).GE.EPS) GO TO 1600

1500 IF(ABS(XI-XXI).GE.EPS) GO TO 1600

1510 CONTINUE

RETURN

1600 IF(IC.GE.ICN) RETURN

BSI=0.0

DO 1700 I=1,N

R(I)=R(I)-ALPHA*V(I)

1700 BSI=BSI+V(I)*R(I)

BETA=BSI/BBO

DO 1800 I=1,N

1800 P(I)=R(I)-BETA*P(I)

GO TO 1111

END

4) 最急降下法およびプログラム

第 0 近似 $X^{(0)} = B$

第 $\nu+1$ 近似 $R^{(\nu)} = AX^{(\nu)} - B$

$P^{(\nu)} = A^T R^{(\nu)}$

$X^{(\nu+1)} = X^{(\nu)} - \frac{R^{(\nu)} P^{(\nu)}}{P^{(\nu)} P^{(\nu)}}$

$.A^T R^{(\nu)}$

SUBROUTINE BOOTH (A,IR,B,X,XX,R,P,V,N,JN,
 ICN,EPS)

DIMENSION A(N,JN),IR(N,JN),B(N),X(N),XX
 (N),R(N),P(N),V(N)

IC=0

DO 1000 I=1,N

1000 X(I)=B(I)

1111 IC=IC+1

DO 1110 I=1,N

XX(I)=X(I)

WS=0.0

DO 1100 K=1,JN

J=IR(I,K)

IF(J.EQ.0) GO TO 1110

1100 WS=WS+A(I,K)*X(J)

1110 R(I)=WS-B(I)

DO 1210 I=1,N

WS=0.0

DO 1200 K=1,JN

J=IR(I,K)

IF(J.EQ.0) GO TO 1210

1200 WS=WS+A(I,K)*R(J)

1210 P(I)=WS

DO 1310 I=1,N

```

WS=0.0
DO 1300 K=1,JN
J=IR(I,K)
IF(J.EQ.0) GO TO 1310
1300 WS=WS+A(I,K)*P(J)
1310 V(I)=WS
VV=0.0
RV=0.0
DO 1400 I=1,N
VV=VV+V(I)*V(I)
1400 RV=RV+R(I)*V(I)
FACTOR=RV/VV
DO 1500 I=1,N
1500 X(I)=XX(I)-FACTOR*P(I)
DO 1610 I=1,N
XI=X(I)
XXI=XX(I)
IF(XI.EQ.0.0) GO TO 1600
IF(ABS(XXI/XI-1.0).GE.EPS) GO TO 1700
1600 IF(ABS(XI-XXI).GE.EPS) GO TO 1700
1610 CONTINUE
RETURN
IF(IC.LT.ICN) GO TO 1111
RETURN
END

```

4. 数値実験

われわれは 3. で述べた 3 通りのプログラムが実際に与えられた行列に対してどのような収束性を持っているか、二、三の数値実験を試みた。

1) 実験に使用した係数行列

この種の実験で最も困難を生ずるのは、一般的な行列を考えられないところにある。しかし、実験する解法の理論的収束条件はつぎのようである。

逐次反復法 $\sum |a_{ij}| < |a_{ii}| (i \neq j)$ のとき適用できる。

共役傾斜法 正値対称行列の場合に適用できる。

最急降下法 無条件に適用できる。

以上の条件より、正値対称であること、非対角元が小さいことを考慮して、次のような行列を与えることにした。

(i) 対角元に非対角元が隣接している場合

$$(a_{i-1}, a_{ii}, a_{i+1})$$

$$(a_{i-j}, \dots, a_{i-1}, a_{ii}, a_{i+1}, \dots, a_{i+j})$$

(ii) 対角元に非対角元が離れている場合

$$(a_{i-j}, 0, \dots, 0, a_{ii}, 0, \dots, 0, a_{i+j})$$

$$(a_{i-k-j}, \dots, a_{i-k}, 0, \dots, 0, a_{ii}, 0, \dots, 0, a_{i+k}, \dots, a_{i+k+j})$$

これらの行列に対して $\sum |a_{ij}| |a_{ii}|$ の値を 10^{-1} , 10^0 , 10^1 とする数値を与えた。これらの数値だけで十分ではないが、解法の特性を検討できる結果を得たので、以下実験結果を報告する。

2) 実験結果

実験は1)で述べたデータに対し、20元、40元、60元、……、300元各元数に対して行なった。また、解の相対誤差が 10^{-4} または残差の二乗平均が 10^{-5} 以下で収束したものとした。この結果、比較的非対角元に対角元に隣接している場合には、収束性は元数の大小にはほとんど関係しないことがわかった。以下これらの実験結果を図示する

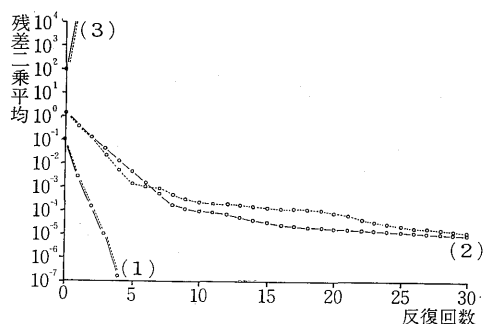


図-2 逐次反復法

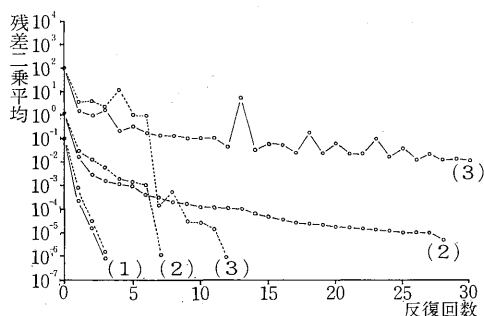


図-3 共役傾斜法

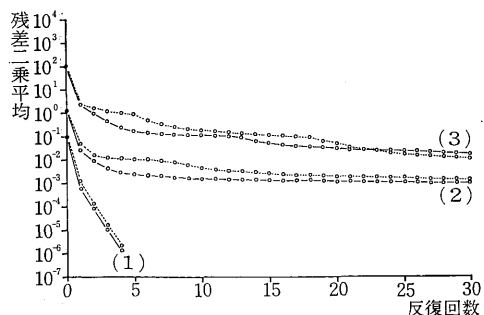


図-4 最急降下法

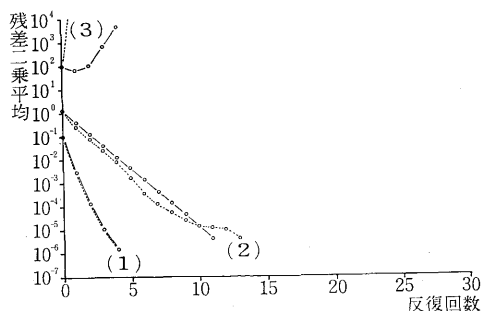


図-5 逐次反復法

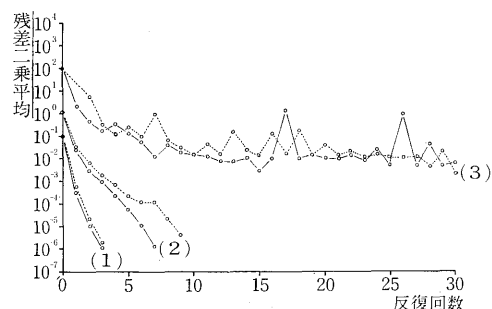


図-6 共役傾斜法

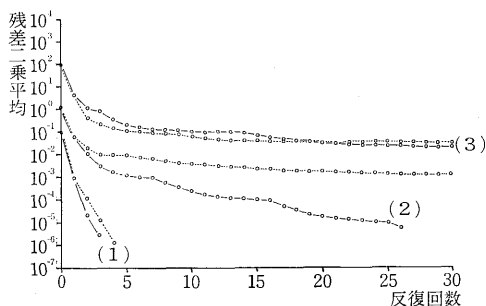


図-7 最急降下法

図-2 から図-7 において、縦軸は解の残差二乗平均の値で、横軸は反復回数を表わしている。また、グラフにおける実線は非対角元が対角元に隣接している場合、破線は比較的離れている場合を表わしている。図-2, 3, 4 は、非零要素が2個の場合、図-5, 6, 7 は、非零要素が10個の場合で、図中(1)の線が $\sum |a_{ij}|/|a_{ii}| = 10^{-1}$ 、(2)の線が $\sum |a_{ij}|/|a_{ii}| = 10^0$ 、(3)の線が $\sum |a_{ij}|/|a_{ii}| = 10^1$ を表わしている

これらの数値以外においても、たとえば、 $\sum |a_{ij}|/|a_{ii}|$ が 10^{-1} と 10^0 の間の値についても実験したが、その結果はすべて(1)と(2)の線の間に入り結局(1)と(2)だけで十分推察できるようである。

次にこの実験をもとに3-1)で述べた解法を比較してみる。

5. 解法の比較およびプログラム使用法

すでに述べたように、連立一次方程式の数値解法は多くあるが、間接法に限って見れば使用できるものはかなり少なくなる。

1) 解法の比較

一般に直接法と間接法との使い分けは、ほとんどの場合、計算時間（必要な精度を得るまでの計算時間）によって選択されている。この時間は元数 n だけで決まり、プログラム中の乗算の回数で大きくばな見当がつけられる。直接法の場合は約 $n^3/3$ 回の乗算を必要とし、間接法の場合は解法によりかなりの差があり、また非零要素の多さにも左右されるが、1回の反復に当たり大体 n^2 回の乗算と見積れば、間接法において $n/3$ 回の反復に要する時間と直接法の時間とが同じくらいになる。このように元数が大きくなれば極端に時間を要するために、間接法で $n/3$ 以下の反復で必要とする精度に達するならば間接法が有利となる。たとえば600元の場合には間接法では大体200回の反復で直接法と同じ程度の時間となるが、実際には200回の反復をするまでもなく必要とする精度に達する場合が多い。このような理由から、現在、大きな行列の場合は間接法を用いている例が多い。そして本文で述べた3つの方法が特によく使用されているようである。

4-1)で述べたように、理論的には収束条件が示されており、この適合条件をあやまらなければ、何らかの解を得ることができる。しかし、これらの解法はそれぞれ特徴ある収束性を示しており、何回の反復でどの程度の精度が得られるのか見当がつかない。理論的には逐次反復法、最急降下法は無限回の反復で真の解となり、共役傾斜法は元数回の反復で真の解となるはずである。しかし計算誤差の累積などで真の解とはならないし、また、これだけ反復させたのでは意味がなくなる。

われわれの実験により正値対称行列に対しては次のことがいえる。

$$(i) \sum |a_{ij}|/|a_{ii}| = 10^{-1} \quad (i \neq j)$$

全て同じ程度の収束性を示している。

$$(ii) \sum |a_{ij}|/|a_{ii}| = 10^0 \quad (i \neq j)$$

どの解法も極端に収束が悪くなるが、数十回の反復で収束する。ただし非対角元より離れている場合は共役傾斜法が最もよく収束している。

$$(iii) \sum |a_{ij}|/|a_{ii}| = 10^1 \quad (i \neq j)$$

共役傾斜法、最急降下法のみ使用できるが、前者は不安定な収束性を示し、後者は安定している。しかし、共役傾斜法は急激に収束する場合があり、特に $\sum |a_{ij}|/|a_{ii}| \quad (i \neq j)$ が 10^0 と 10^1 の

間の値のときは有利のように思う。

これらの実験は 3-2), 3), 4) に示したプログラムを使用した。このプログラムのリンクの仕方を次に述べておく。

2) プログラム使用法

3-2), 3), 4) のプログラムを使用するためには、次の条件を満足していなければならない。

(i) プログラム中の変数名 A, IR は使用前にクリアしておくこと。

(ii) 係数行列は図-1 のように変換し、対角元 a_{ii} で正規化しておくこと。

以上の処理をしておけば使用できる。またこのプログラム中の仮の引数のもつ意味は次のとおりである。

A	係数行列で実数型 2 次元配列。
IR	非零要素の列の位置を示すデータで整数型 2 次元配列。
B	右辺定数ベクトルで実数型 1 次元配列。
X	解ベクトルで実数型 1 次元配列。
XX, R, P, V	作業ベクトルで実数型 1 次元配列。
N	方程式の元数で整数型変定数。
JN	A の最大列番号で整数型変定数。

EPS 相対誤差判定値で実数型変定数。

ICN 最大反復回数で整数型変定数。

IC 主プログラムに戻る前の反復回数で整数型変数。

与えられた各データは全て保存されている。なおこれらのプログラムの他に、FACOM 270-30 で内蔵ドラムを使用して 1000 元まで処理できるプログラムも開発しているので使用されたい。

6. あとがき

以上のように、われわれは中部工業大学電子計算機センター・ライブラリ・プログラム開発とあわせて解法の検討を行なっている。連立一次方程式に関するものだけでもまだ端緒についたばかりでもあるが、今後直接法も含めて検討する予定である。なおこれらのプログラムのユーザーズマニュアルは計算機センターにあるので参照されたい。

最後に、本報告にあたり終始適切な助言をいただいた本学土木工学科の桑山八洲彦講師、ならびに、名古屋大学工学部大学院博士課程の梶田建夫氏に感謝の意を表します。