7K－2

帯行列を係数行列とする連立一次方程式を誤差なく計算する場合のＬＵ分解について

大柳　俊夫　　大内　東
（北海道大学工学部）

１．はじめに

数値計算法の中で，基本的かつ重要なものの一つである連立一次方程式を解く方法に対する要求は，コンピュータ支援の下で大規模な問題を高速かつ高精度に解くことである．そのために，問題の係数行列の構造や性質を生かしたアルゴリズム，そのアルゴリズムの性能を十分に発揮するような計算機上へのインプリメント方法などが研究されている[1),2)]．最近では，計算機のハードウェアの処理能力の急速な向上に伴い，その性能をフルに引出すためのアルゴリズムおよびインプリメント方法の見直しが行われている．例えば，スパーコンピュータを使った大規模計算や高性能のパソコン／ワークステーション上での数式処理と数値計算を融合した計算に関する研究が盛んに行われている[2)]．

本報告では，有理数演算を手軽に使用できる数式処理システムを用いて，係数行列が帯状である連立一次方程式を誤差なく効率良く解くためのＬＵ分解の方法を検討する．連立一次方程式をＬＵ分解して解く場合は，四則演算と比較演算のみで計算を行うことが可能であるので有理数演算を用いることができる．なお，数式処理システムとしてはREDUCEを用いた．

２．有理数演算による計算の振舞い

有理数演算を用いて，帯行列を係数行列とする連立一次方程式をＬＵ分解により解く場合の振舞いについては，ＬＵ分解する際のピボット選択の違いが使用メモリー容量や計算時間へ及ぼす影響が大きく，フィル・インの多く生じるピボット選択の方が，フィル・インのほとんど生じないピボット選択よりも使用メモリー容量と計算時間の点で有利な場合があることが確認されている[3),4)]．この特徴は，浮動小数点演算の場合にはまず起こり得ないことで，有理数演算を用いる場合に特有のことと考えられる．このような現象の生じる理由は，浮動小数点演算の場合はある浮動小数点数を表すために必要なメモリー容量が計算途中一定であるのに対し，有理数演算の場合は計算途中で分子分母の中間膨張が起こることがあり，そのため多くのメモリーと多くの計算時間が必要になる．そして，フィル・インが生じないかわりに分子分母の桁数の大きな数が多く現れ，結局フィル・インが生じても桁数の大きな数が少なくなるようなピボット選択の方が，使用メモリー容量と計算時間の両方の点で有利になる．この状況は，ＬＵ分解後の行列ＬＵの構造とその分子分母の桁数の和(65536進数) を調べた図１，２の結果から明らかである[4)]．図１は，係数行列の第ｋ列成分をa_{ik}とすると，

$$p = \min_{k \le i \le n} \{ i \mid a_{ik} \neq 0 \} \qquad (1)$$

に対応する要素a_{pk}を第ｋ段階目の分解のピボットとした結果，図２はa_{ik}の分子分母の桁数の和をS_iとするとき

$$S_p = \min_{k \le i \le n} \{ S_i \mid a_{ik} \neq 0 \} \qquad (2)$$

に対応する要素a_{pk}をピボットとした結果である．なおS_iの計算は，有理数演算を行うためのライブラリーを独自に実現して利用したので，容易に行うことができた．また実験で用いた係数行列は，その次数ｎを50とし，成分a_{ij}を

$$1 \le a_{ij} \le 5 \quad (1 \le i, j \le n, |i-j| \le 2) \qquad (3)$$

の範囲の整数乱数とした．

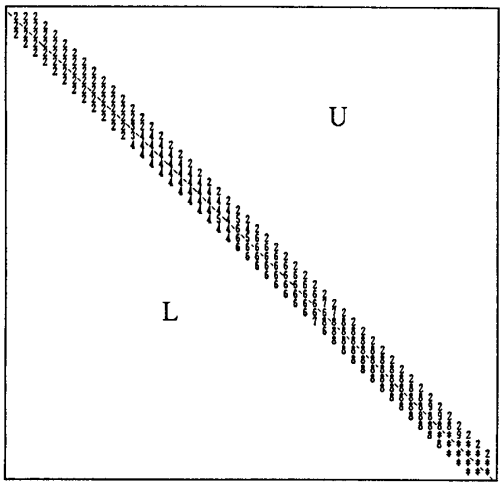

図１．(1)式を用いた場合の行列ＬＵの構造

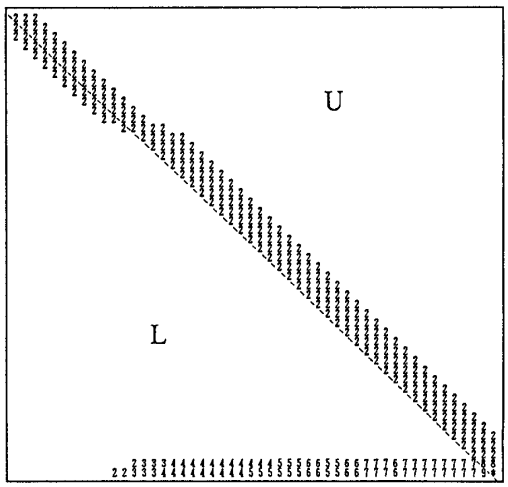

図２．(2)式を用いた場合の行列ＬＵの構造

On LU Decomposition for an Exact Solution of a Linear Equations with Band Matrix

Toshio OHYANAGI and Azuma OHUCHI
HOKKAIDO UNIVERSITY

3．新しいピボット選択とＬＵ分解の方法

図１，２より，分子分母の桁数の大きな数をできる限り少なくするようなピボット選択方法が有効で，(2) 式はそのようなピボット選択の一つと考えられる．しかし，REDUCEで分子分母の桁数を求めることはそれほど容易ではなく，もっと簡単なピボット選択方法もしくはＬＵ分解方法を考案しなければならない．そこで，ＬＵ分解した結果が図２に示す構造に近ければ良いと考え，そのような構造にする方法を２つ考案した．第１の方法は，

$$p = \mathbf{max}_{k \le i \le n} \{ i \mid a_{ik} \neq 0 \} \qquad (4)$$

に対応する要素a_{pk}を第ｋ段階目の分解のピボットとするものである．この方法を用いて２章で述べた実験と同じ実験を行った結果を図３に示す．また，第２の方法はピボット選択を行わず直接係数行列を図３の形式にＬＵ分解する方法で，そのアルゴリズムは以下の通りである．

```
nm:=n-m;
for k:=1 to nm do P[k]:=k+m;
for k:=nm+1 to n do P[k]:=k-nm;
for i:=1 to m do begin
  for j:=1 to nm+i-1 do begin
    sum:=0;
    for k:=1 to j-1 do
      sum:=sum+a[P[nm+i],k]*a[P[k],j];
    a[P[nm+i],j]:=(a[P[nm+i],j]-sum)/
                  a[P[nm+i],j];
  end;
  for j := nm+i to n do begin
    sum:=0;
    for k:=1 to nm+i-1 do begin
      sum:=sum+a[P[nm+i],k]*a[P[k],j];
    a[P[nm+i],j]:=a[P[nm+i],j]-sum
  end;
  for i:=1 to n do
    a[P[i],i]:=1/a[P[i],i]
end;
```

ここで，nは行列の次数，mは半帯幅，Ｐは行置換のための配列である．このアルゴリズムは，ＬＵ分解する以前の係数行列が，

条件：i-j=m を満たす全てのa_{ij}が非零

を少なくとも満足していなければ用いることはできない．この方法を用いて２章で述べた実験と同じ実験を行った結果を図４に示す．

4．数値実験と結果

ＬＵ分解を行う方法として，①ピボット選択を(2)式，②ピボット選択を(4)式，③ピボット選択を行わない，という３つの方法に対して，ＬＵ分解に要する時間と分解後の計算に要する時間を調べる実験を行った．なお実験では，以下に示すＩの形式の問題に対してn=50，100，ＩＩに対してn=50の問題をそれぞれ５題づつ解いた．実験結果を表１に示す．表中の値は問題５題を解いた結果の平均値であり，この結果，①よりも②および③が有効であることがわかる．

Ｉ：a_{ij}を$1 \le a_{ij} \le 20$の範囲の整数乱数
$(1 \le i, j \le n, |i-j| \le 2)$．

ＩＩ：a_{ij}を$1 \le a_{ij} \le 10000$の範囲でランダムに発生させた素数$(1 \le i, j \le 50, |i-j| \le 2)$．

5．おわりに

本報告では，有理数演算を用いて，係数行列が帯状である連立一次方程式を誤差なく効率良く解くためのＬＵ分解の方法を提案し，数値実験によりその有効性を示した．

参考文献

1)森，名取，鳥居：数値計算，岩波書店，東京(1982)．

2)津田：数値処理プログラミング，岩波書店，東京(1988)．

3)大柳，大内：“有理数演算を用いた連立一次方程式の計算”，第15回システムシンポジウム・第10回知識工学シンポジウム合同シンポジウム講演論文集，p209-214，札幌(1989)

4)大柳，大内：“有理数演算を用いた線形方程式の解法の振舞い”，情報処理学会研究報告（数値解析）

表１．ＬＵ分解方法の違いに関する実験結果

		ＬＵ分解に要した時間	ＬＵ分解後の計算時間	合計時間
Ｉ	①	45.4(s)	66.3(s)	111.7(s)
(n=50)	②	42.4	54.5	96.9
	③	42.3	49.6	91.9
Ｉ	①	203.3	351.4	554.7
(n=100)	②	161.2	257.5	418.7
	③	159.9	220.5	380.4
ＩＩ	①	226.3	199.8	426.1
(n=50)	②	196.8	99.6	296.4
	③	198.3	93.7	292.0

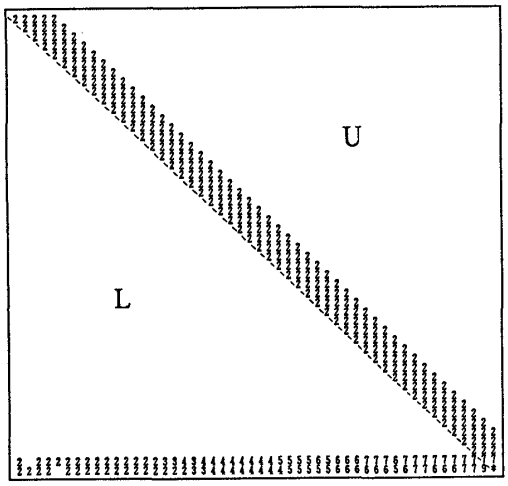

図３．(4)式を用いた場合の行列ＬＵの構造

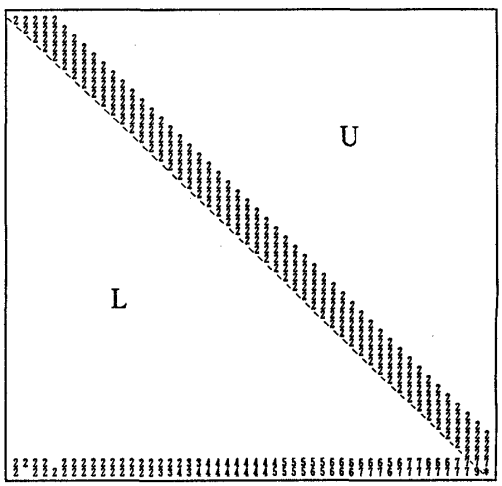

図４．直接ＬＵ分解した場合の行列ＬＵの構造