

線形代数(I 逆行列)

—数値計算(07/10/26)—

関西学院大学理工学部 西谷滋人

Copyright ©2007 by Shigeto R. Nishitani

行列計算

数値計算の中心課題の一つである、行列に関する演算について見ていく。多次元、大規模な行列に対する効率のよい計算法が多数開発されており、多くの既存のライブラリが用意されている。本章ではそれらの中心をなす、**逆行列(matrix inverse)**と**固有値(Eigen values)**に関して具体的な計算方法を示す。現実的な問題には既存のライブラリを使うのが上策であるが、それでも基礎となる原理の理解や、ちょっとした計算、ライブラリの結果の検証に使えるルーチンを示す。

逆行列は**連立一次方程式**を解くことと等価である。ルーチン的なやり方に**ガウスの消去法**がある。これは上三角行列になれば代入を適宜おこなうことで解が容易に求まることを利用する。さらに、初期値から始めて次々に解に近づけていく反復法がある。この代表例である**Jacobi(ヤコビ)法**と、収束性を高めた**Gauss-Seidel(ガウス-ザイデル)法**を紹介する。

上記の手法をより高速にした修正コレスキー分解と共役傾斜(共役勾配)法があるが、少し複雑になるので割愛する。必要ならばNumRecipeを読め。

ガウス消去法による連立一次方程式の解

原理と手作業

逆行列は連立一次方程式を解くことと等価である。すなわち、**A**を行列、**x**を未知数ベクトル、**b**を数値ベクトルとすると、

$$\begin{aligned}Ax &= b \\ A^{-1}x &= A^{-1}b \\ x &= A^{-1}b\end{aligned}$$

である。未知数の少ない連立一次方程式では、適当に組み合わせて未知数を消していけばいいが、未知数が多くなってしまうと破綻する。未知数の多い多元連立一次方程式で、ルーチン的に解を求めていく方法がガウス消去法で、前進消去と後退代入という2つの操作からなる。

後退代入(Backward substitution)による解の求め方を先ず見よう。たとえば、

$$\begin{aligned}x+y-2z &= -4 \\ -3y+3z &= 9\end{aligned}$$

$$-z=-2$$

では、下から順番に $z \rightarrow y \rightarrow x$ と適宜代入することによって、簡単に解を求めることが出来る。係数で作る行列でこのような形をした上三角行列にする操作を**前進消去**あるいは**ガウスの消去法(Gaussian elimination)**という。下三角行列L(lower triangular matrix)と

上三角行列U(upper triangular matrix)の積に分解する操作

$$A = LU$$

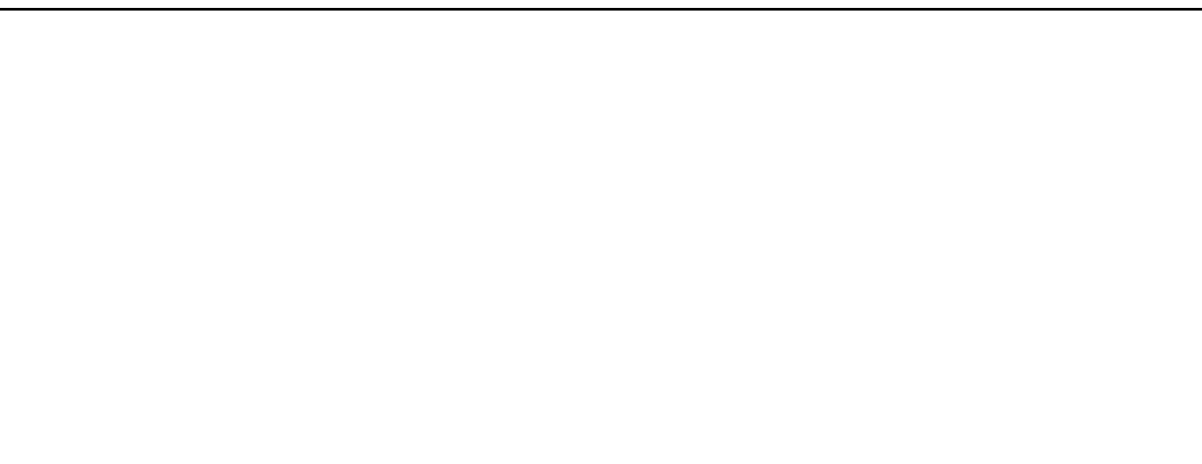
を**LU分解(LU decomposition)**という。例えば先に示した上三角行列を係数とする連立方程式は、

$$x+y-2z=-4$$

$$x-2y+z=5$$

$$2x-2y-z=2$$

を変形することで得られる。この係数行列のU行列が先程の(¥ref{Eq:U})式の係数行列である。また、このL行列は



となる。

▼ MapleによるLU分解

係数行列(coefficient matrix)と定数項(b)との関係は以下の通りである。

```
> restart;
A:=Matrix([[1,1,-2],[1,-2,1],[2,-2,-1]]):
X:=Vector([x,y,z]):
#X:=Vector([1,-1,2]):
b:=Vector([-4,5,2]):
A.X=b;
```

$$\begin{bmatrix} x+y-2z \\ x-2y+z \\ 2x-2y-z \end{bmatrix} = \begin{bmatrix} -4 \\ 5 \\ 2 \end{bmatrix}$$

単に逆行列を求める際は

```
> with(LinearAlgebra):
MatrixInverse(A);
```

$$\begin{bmatrix} \frac{4}{3} & \frac{5}{3} & -1 \\ 1 & 1 & -1 \\ \frac{2}{3} & \frac{4}{3} & -1 \end{bmatrix} \quad (3.2.1)$$

である。Mapleでは行列を三角行列に分解するために、LUDecompositionコマンドが用意されている。係数と定数項から作られる行列を 拡大係数行列(augmented matrix) といい $\text{verb} \langle A|b \rangle$ のようにして作られる。

```
> P,L,U:=LUDecomposition(<A|b>);
```

$$P, L, U := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & \frac{4}{3} & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & -2 & -4 \\ 0 & -3 & 3 & 9 \\ 0 & 0 & -1 & -2 \end{bmatrix}$$

後退代入までおこなってもとめた連立方程式の解は以下の通り。

```
> LUDecomposition(<A|b>, output='R');
```

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \end{bmatrix} \quad (3.2.2)$$

▼ MapleによるLU分解のプログラム

LU分解すれば線形方程式の解が容易に求まることは理解できると思う。具体的に **A** をLU分解する行列(消去行列と称す) **T1**, **T2** の係数は次のようにして求められる。

```
> A0:=Matrix([[1,1,-2],[1,-2,1],[2,-2,-1]]):
b0:=Vector([-4,5,2]):
A:=Matrix(A0):
B:=Vector(b0):
n:=3:
L:=Matrix(array(1..n,1..n,identity)):
for i from 1 to n do
T[i]:=Matrix(array(1..n,1..n,identity)):
for j from i+1 to n do
am:=A[j,i]/A[i,i]:
T[i][j,i]:=-am:
L[j,i]:=am:
for k from 1 to n do
A[j,k]:=A[j,k]-am*A[i,k]:
end do:
B[j]:=B[j]-B[i]*am:
end do:
end do:

> T[1];
T[2];
```

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{4}{3} & 1 \end{bmatrix}$$

```
> U:=T[2].T[1].A0;
```

$$U:=\begin{bmatrix} 1 & 1 & -2 \\ 0 & -3 & 3 \\ 0 & 0 & -1 \end{bmatrix}$$

```
> L.U;
```

$$\begin{bmatrix} 1 & 1 & -2 \\ 1 & -2 & 1 \\ 2 & -2 & -1 \end{bmatrix}$$

(3.3.1)

```
> L;A;
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & \frac{4}{3} & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & -2 \\ 0 & -3 & 3 \\ 0 & 0 & -1 \end{bmatrix}$$

(3.3.2)

```
> B;
```

$$\begin{bmatrix} -4 \\ 9 \\ -2 \end{bmatrix}$$

▼ ピボット操作

ガウス消去法で困ることに、もしある行を対角要素で割ろうとするとときに、その対角要素が0になる場合がある。しかし、この場合にも、方程式の順序(行列の行と右辺の値)を入れ替えれば解決する。ちなみにこの割るほうの要素を**ピボット要素**あるいは**ピボット (pivot)** (バスケットの軸足を動かさずにくるくる回すやつ)と呼ぶ。この操作は変数の並びは変わらず単に方程式の順番を変更しただけである。

さらに対角要素の数値が厳密に0でなくとも極端に0に近づいた場合でも、その数で

割った数値が大きくなり他の数との差を取ると以前に示した情報落ちの可能性が出てくる。この現象を防ぐためには、絶対値最大のピボットを選んで行の入れ替えを毎回おこなうといい結果が得られることが知られている。

Mapleの**LUdecomposition**コマンドをこのような行列に適用すると、**置換行列(permutation matrix)****P**は単位行列ではなくなる。 **$P \cdot A = L \cdot U$** であることに注意。

▼ 反復法による連立方程式の解

以下のような連立方程式を

$$\begin{bmatrix} 5x + y + z + u \\ x + 3y + z + u \\ x - 2y - 9z + u \\ x + 3y - 2z + 5u \end{bmatrix} = \begin{bmatrix} -6 \\ 2 \\ -7 \\ 3 \end{bmatrix}$$

形式的に解くと

$$x = \frac{-6 - (x + y + z)}{5}$$

$y =$

$z =$

$u =$

となる。適当に初期値(x_0, y_0, z_0, u_0)をとり、下側の方程式に代入すると、得られた出力(x_1, y_1, z_1, u_1)はより正解に近い値となる。これを繰り返すことによって正解が得られる。これを**ヤコビ(Jacobi)法**と呼び、係数行列の対角要素が非対角要素にくらべて大きいときに適用できる。多くの現実の問題ではこの状況が成り立っている。

Gauss-Seidel法はJacobi法の高速版である。n番目の解の組が得られた後に一度に次の解の組に入れ替えるのではなく、得られた解を順次改良した解として使っていく。これにより、収束が早まる。以下にはヤコビ法のコードを示した。x1[i]の配列を変数に換えるだけで、Gauss-Seidel法となる。

```
> AA:=Matrix([[5,1,1,1],[1,3,1,1],[1,-2,-9,1],[1,3,-2,5]]):  
b:=Vector([-6,2,-7,3]):  
n:=4;  
x0:=[0,0,0,0]:
```

```

x1:=[0,0,0,0]:
for iter from 1 to 20 do
for i from 1 to n do
  x1[i]:=b[i];
  for j from 1 to n do
    x1[i]:=x1[i]-AA[i,j]*x0[j];
  end do:
  x1[i]:=x1[i]+AA[i,i]*x0[i];
  x1[i]:=x1[i]/AA[i,i];
end do:
x0:=evalf(x1);
print(iter,x0);
end do:

```

$n := 4$

```

1, [-1.200000000, 0.6666666667, 0.7777777778, 0.6000000000]
2, [-1.608888889, 0.6074074073, 0.5629629630, 0.7511111112]
3, [-1.584296296, 0.7649382717, 0.5474897119, 0.7825185186]
4, [-1.618989300, 0.7514293553, 0.5187050756, 0.6768921810]
5, [-1.589405322, 0.8077973477, 0.5061160189, 0.6804222770]
6, [-1.598867129, 0.8009556753, 0.4972691400, 0.6356490634]
7, [-1.586774776, 0.8219829753, 0.4927633981, 0.6381076766]
8, [-1.590570810, 0.8186345670, 0.4897074389, 0.6212705292]
9, [-1.585922507, 0.8265309473, 0.4881589539, 0.6228163974]
10, [-1.587501260, 0.8249823853, 0.4870924439, 0.6165295146]
11, [-1.585720869, 0.8279597673, 0.4865626093, 0.6173477984]
12, [-1.586374035, 0.8272701537, 0.4861897104, 0.6149933572]
13, [-1.585690644, 0.8283969890, 0.4860087794, 0.6153885990]
14, [-1.585958873, 0.8280977553, 0.4858782197, 0.6145034472]
15, [-1.585695884, 0.8285257353, 0.4858165626, 0.6146844092]
16, [-1.585805341, 0.8283983040, 0.4857707838, 0.6143503606]
17, [-1.585703890, 0.8285613990, 0.4857498236, 0.6144303994]
18, [-1.585748324, 0.8285078890, 0.4857337457, 0.6143038680]
19, [-1.585709101, 0.8285702367, 0.4857266407, 0.6143384296]
20, [-1.585727061, 0.8285480103, 0.4857209840, 0.6142903344]

```

(4.1)

課題

1 後退代入法で解を求めよ.

$$x+4y-3z=1$$

$$-6y+4z=1$$

$$-5/3z=1/3$$

2 次の行列AをLU分解せよ.

```
> Matrix([ [1,4,3], [1,-2,1], [2,-2,-1] ]);
```

$$\begin{bmatrix} 1 & 4 & 3 \\ 1 & -2 & 1 \\ 2 & -2 & -1 \end{bmatrix} \quad (5.1)$$

3 Jacobi法のプログラムを参照してGauss-Seidel法のプログラムを作れ。Jacobi法と収束性を比べよ。

4 次の連立方程式の解を求めよ。ただし、pivot操作が必要となる。次の連立方程式の解を求めよ。ただし、pivot操作が必要となる。

```
> with(LinearAlgebra):
A:=Matrix([ [3,2,2,1], [3,2,3,1], [1,-2,-3,1], [5,3,-2,5] ]):
X:=Vector([w,x,y,z]):
b:=Vector([-6,2,-9,2]):
A.X=b;
```

$$\begin{bmatrix} 3w + 2x + 2y + z \\ 3w + 2x + 3y + z \\ w - 2x - 3y + z \\ 5w + 3x - 2y + 5z \end{bmatrix} = \begin{bmatrix} -6 \\ 2 \\ -9 \\ 2 \end{bmatrix} \quad (5.2)$$

5 pivot操作を含めたLU分解のプログラムを作成せよ。上の問題を解き、その\$L\$, \$U\$行列および\$L^{-1} \cdot b\$ベクトルを求めよ。