

コレスキー分解

コレスキー分解を行う関数 `CholeskyDecom` をヘッダーファイル `mymat.h` に以下のように宣言した。

```
int CholeskyDecom( myMat a, myMat & b );
```

この関数の使用例をリスト 1 に示す。

リスト 1 コレスキー分解。

```
#include "stdafx.h"
#include <math.h>
#include "mymat.h"
#include "smpIRN.h"

using namespace System;
using namespace myLib;
using namespace smpIRN;

int main(array<System::String ^> ^args)
{
    smpIRN RN;
    cout << "n = ";
    int n;
    cin >> n;
    myMat b(n, 2*n);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < 2*n; j++)
            b[i][j] = RN.uni();
    cout << "b = " << endl;
    WriteMat( 15, b );
    myMat a(n, n);
    a = b * matTransp(b);
    cout << "a = " << endl;
    WriteMat( 15, a );
    myMat L(n, n);
    int ick = CholeskyDecom( a, L );
    if (ick != 0) {
        cout << "CholeskyDecom-Error ! ... ick = " << ick << endl;
    } else {
        cout << "L = " << endl;
        WriteMat( 15, L );
        cout << "a - L*L' = " << endl;
        WriteMat( 15, a - L * matTransp(L) );
    }
    Console::WriteLine("Enter キーを押して終了");
    Console::ReadLine();
    return 0;
}
```

リスト 1 では、正定値対称行列 a を、一様乱数によって要素の値を設定した行列 b から

$$a = bb'$$

によって得ている。一様乱数は、ヘッダーファイル `smplrN.h` の名前空間 `smplrN` に宣言されているクラス型 `smplrN` のメソッド `uni()` の呼び出して生成している。

リスト 1 において `#include` の対象となっているヘッダーファイル `mymat.h` と `smplrN.h` を利用するときは、それらのファイルで宣言されているものの定義が置かれているファイル `mymat.cpp` と `smplrN.cpp` も他の `*.cpp` ファイルや `*.h` ファイルと同じフォルダにコピーして、メニュー「プロジェクト | 既存項目の追加」により追加しておく。ヘッダーファイルは `#include` により挿入されているが、`*.cpp` ファイルはこの「プロジェクト | 既存項目の追加」によって追加しておかないとビルド時にエラーとなる。

リスト 1 のプログラムの実行例を図 1 に示す。

```
n = 3
b =
  1.60818e-005    0.110741    0.76308    0.179978    0.902878
    0.88984
  0.387011    0.475943    0.882193    0.18578    0.6388
  0.269213
  0.246683    0.136171    0.183017    0.784487    0.716519
  0.279469
a =
  2.23395    1.57565    1.19154
  1.57565    1.66962    1.00043
  1.19154    1.00043    1.31981
L =
  1.49464    0    0
  1.0542    0.747185    0
  0.79721    0.214148    0.799005
a · L*L' =
-4.44089e-016 -2.22045e-016    0
-2.22045e-016    0    0
    0    0    0
Enter キーを押して終了
```

図 1 リスト 1 のプログラムの実行例。

プログラムを実行すると「 $n =$ 」と行列の大きさを聞いてくるので、適当なサイズを設定する。Enter キーを押すと、計算が始まり、最後に行列 a のコレスキー分解 $a = LL'$ の精度 $a - LL'$ が「 $a - L*L' =$ 」の値として表示される。図 1 では `double` 型の有効桁数の範囲で 0 であることがわかる。