LAPACK と線形計算

SHIMURA　Masato
homepage3.nifty.com/asagaya_avenue

2015 年 5 月 14 日

はじめに

J は Ver.8 で GUI に QT を採用し、更に J803 からは JEngine を改定した。GUI を用いないスクリプトには大きな変更はないが、細部は確認が必要である。

LAPACK 関連も大きな変更はないが細かな改良は続けられている。最新の線形計算パッケージ LAPACK のテストを兼ね LAPACK のパワーエクササイズを行い、併せて他の線形計算ツールとの照合も行ってみよう。

1　LAPACK

LAPACK　*Linear Algebra Package*
定評ある *FORTRAN* の線形計算パックに *J* 用の *DLL* と *J* のスクリプトを整備し、主要部分を *J* から利用できるようにしたもの。
[*1]

利用できる線形計算関数　豊富な *LAPACK* の関数のうち次の線形計算関数が *J* で利用できる

- 実数の固有値と固有ベクトル　　*geev*
- 複素数の固有値と固有ベクトル　　*geev*
- エルミート行列の固有値と固有ベクトル　　*heev*
- *SVD* 分解　　*gesvd*

[*1] この LAPACK は C で記述され、FORTRAN にコード変換されるようだ。

- コレスキー分解　　*potrf*
- *QR* 分解　　*geqrf*
- *LU* 分解　　*getrf*

*2

lapack の準備　*LAPACK* の使用に際しての *2* つの事前準備

1. *jpath* を通す。次の *1* 行をタイプする

```
   jpath '~addons/math/lapack'

c:/language/j64-803/addons/math/lapack
```

2. *lapack.ijs* のロード（共通）

```
 load 'math/lapack'
```

2　固有値を求める

関数のロード　実数/複素数の固有値計算関数のロード

```
load 'math/lapack/geev'
```

例 1:　非対称行列 (実数解）

- 例題の入力。行ごとにカンマ (,) でつなぎ、最終行はラミネート (,:) で連結しマトリクスにくみ上げる。

```
   A0=: 2 _2 3,1 1 1,:1 3 _1
   A0
2 _2  3
1  1  1
1  3 _1
```

- 固有値と固有ベクトルの計算

```
   ] a0=.  geev_jlapack_ A0
+-----------------------------+-------+----------------------------+
|0.771517  0.486664         0|3 1 _2|0.57735  0.57735   _0.616849|
|0.154303 _0.811107 _0.707107|       |0.57735 _0.57735 _0.0560772|
|0.617213  0.324443  0.707107|       |0.57735 _0.57735   0.785081|
```

*2 古いバージョンの J に同梱の LAPACK では *dgeev* のように前に **d** が付いていた。*geev* は *dgeev* でも再定義されており、今でも通る。

```
+--------------------------+------+-------------------------+
```

固有値　固有ベクトル

- 対角化

$$P^{-1}AP = D = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_3 \end{pmatrix}$$

```
   ] p=.  >2{a0
0.57735  0.57735  _0.616849
0.57735 _0.57735 _0.0560772
0.57735 _0.57735   0.785081


 clean  (%. p) +/ . * A1  +/ . *  p

3 0  0
0 1  0
0 0 _2
```

*3

例 2:　非対称行列 (複素数を含む解）

- 固有値と固有ベクトル

```
   ] A1=: 0 3 4,0 _1 _2,: _1 2 4
 0  3  4
 0 _1 _2
_1  2  4
   ],. a1=. geev_jlapack_ A1
+---------------------------------------------+
|_0.353553j_0.353553 _0.353553j0.353553  0.666667|
| 0.353553j_0.353553  0.353553j0.353553  0.333333|
|           0.707107           0.707107 _0.666667|
+---------------------------------------------+
```

*3 数値計算の塵取り関数 *clean* を含む *numeric.ijs* は *lapack.ijs* でロードされている

```
|1j1 1j_1 1                                     |NB. 固有値
+-----------------------------------------------+
|          0.725476            0.725476 _0.57735 |NB. 固有ベク
トル
|_0.435286j0.145095 _0.435286j_0.145095  0.57735 |
|0.507833j0.0725476 0.507833j_0.0725476 _0.57735 |
+-----------------------------------------------+
```

- 対角化

```
   p1=.> 2{ a1
   clean  (%. p1) +/ . * A1 +/ . *  p1
1j1    0 0
  0 1j_1 0
  0    0 1
```

例 3: ジョルダン標準形は作れるか

- 固有値は 5 重根を持つ

```
   ] T0=: 2 1 0 0 0 0,_1 4 0 0 0 0,_1 1 2 1 0 0,_1 1 _1 4 0 0,_1 1 _1 1 3
 2 1  0 0 0 0
_1 4  0 0 0 0
_1 1  2 1 0 0
_1 1 _1 4 0 0
_1 1 _1 1 3 0
_1 1 _1 1 1 2
   ],. t0=. geev_jlapack_ T0
+-------------------------------------------------------------+
|        0    0.456958  0.456958  0.256288  0.707107  0.707107|
|        0   _0.456958 _0.456958 _0.256288 _0.707107 _0.707107|
|        0     0.53962   0.53962 _0.659027         0         0|
|        0    _0.53962  _0.53962  0.659027         0         0|
|_0.707107 2.90895e_16         0         0         0         0|
| 0.707107           0         0         0         0         0|
```

```
+--------------------------------------------------------------+
|2  3  3  3  3  3                                              |
+--------------------------------------------------------------+
|0        0        0         0 0.679782  0.42167               |
|0        0        0         0 0.679782  0.42167               |
|0        0 0.688247 _0.444822 0.122461 0.357075               |
|0        0 0.688247 _0.444822 0.122461 0.357075               |
|0 0.707107 0.162221 _0.549667 0.151325 0.441239               |
|1 0.707107 0.162221 _0.549667 0.151325 0.441239               |
+--------------------------------------------------------------+
```

- 固有ベクトルの行列式の値は 0 になる

```
   ] t1=.> 2{ t0

   clean   -/ . * t1
0
```

- 対角化はできない。ジョルダン標準形も現れない

```
       clean   (%.t1) +/ .*T0+/ .*t1
|domain error
```

3 エルミート行列

エルミート行列には *heev.ijs*} が用意されている

```
 load 'math/lapack/heev'

 ] E=: 2 3j_3,: 3j3 5
 2 3j_3
3j3    5
```

- エルミート行列は ${}^{t}A = A^{*}$ を満たしている行列 **A** である

```
   + |: E    NB. |: is transpose  and  + is conjugate
  2 3j_3
```

```
3j3    5
```

- 固有値と固有ベクトル *(clean* で複素数の塵を取った）

```
 e0=.  heev_jlapack_ E
+----+----------------------------------+
|_1 8|0.57735j_0.57735 0.408248j_0.408248|
|    |        _0.57735          0.816497|
+----+----------------------------------+
```

- 対角化にはユニタリ行列を用いる

```
       ] u0=. >1{e0
0.57735j_0.57735 0.408248j_0.408248
        _0.57735          0.816497
```

- ユニタリ行列 ${}^tU^*U = E$

```
   clean (|: u0) +/ . * + u0
1 0
0 1
```

- 対角化

```
   clean (%. u0) +/ . * E +/ . * u0
_1 0
 0 8
```

4 SVD

SVD singular value decomposition

1. *jlapack.dll* のロード（共通なので一度でよい）

   load 'math/lapack'

2. *SVD* 計算関数のロード

   load 'math/lapack/gesvd'

```
   S=: 3 2 2,2 7 2,: 2 2 5
   S
3 2 2
2 7 2
2 2 5


   geev_jlapack_ A
+-----------------------------+-------+----------------------------+
|0.771517  0.486664         0|3 1 _2|0.57735  0.57735   _0.616849|
|0.154303 _0.811107 _0.707107|       |0.57735 _0.57735 _0.0560772|
|0.617213  0.324443  0.707107|       |0.57735 _0.57735   0.785081|
+-----------------------------+-------+----------------------------+


   gesvd_jlapack_     S
+-----------------------------+-----------------------+-----------------------------+
|_0.397113  0.233192 _0.88765|9.42864       0       0|_0.397113  0.233192 _0.88765|
|_0.755789 _0.631781 0.172148|      0 3.92162       0|_0.755789 _0.631781 0.172148|
|_0.520657  0.739239 0.427132|      0       0 1.64974|_0.520657  0.739239 0.427132|
+-----------------------------+-----------------------+-----------------------------+
    U                                   D                       V
```

5　コレスキー分解

- 関数のロード

  load 'math/lapack/potrf'

- 数値例

```
   ] C=: 4 2 _1,2 4 1,: _1 1 4
  4 2 _1
  2 4  1
_1 1  4
```

- 計算

```
   potrf_jlapack_ C
   2       0       0
   1  1.73205       0
_0.5 0.866025 1.73205
```

- コレスキー分解の数式と計算過程

$$A = LL^T = \begin{pmatrix} l_{11} & & & & & \\ l_{21} & l_{22} & & & & \\ l_{31} & l_{32} & l_{33} & & & \\ & \cdot & & \cdot & & \\ & \cdot & & & \cdot & \\ l_{n1} & l_{n2} & l_{n3} & \cdot & \cdot & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & & & l_{n1} \\ & l_{22} & l_{32} & \cdot & \cdot & l_{n2} \\ & & l_{33} & & & l_{n3} \\ & & & \cdot & & \cdot \\ & & & \cdot & \cdot & \\ & & & & & l_{nn} \end{pmatrix}$$

$$\begin{aligned} l_{11} &= \sqrt{a_{11}} &&= \sqrt{4} &&= 2 \\ l_{21} &= \frac{a_{21}}{l_{11}} &&= 1 \\ l_{31} &= \frac{a_{31}}{l_{11}} &&= -\frac{1}{2} \end{aligned}$$

$$\begin{aligned} l_{22} &= \sqrt{a_{22} - l_{21}^2} &&= \sqrt{4-1} &&= \sqrt{3} \\ l_{32} &= \frac{a_{32} - l_{31} l_{21}}{l_{22}} &&= \frac{1 + 1r2}{\sqrt{3}} &&= \frac{\sqrt{3}}{2} \\ l_{33} &= \sqrt{a_{33} - l_{31}^2 - l_{32}^2} &&= \sqrt{4 - 1r4 - 3r4} = \sqrt{3} \end{aligned}$$

- まとめ

$$\begin{pmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & \sqrt{3} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & \sqrt{3} \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 1 & \sqrt{3} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & \sqrt{3} \end{pmatrix}^T$$

6 LU 分解

- 関数のロード

```
load 'math/lapack'
load 'math/lapack/getrf'
```

- 例

```
  ] A=: 1 4 3,2 7 9,:5 8 _2
1 4  3
2 7  9
5 8 _2
```

- *LU* 分解

```
      ] a=. getrf_jlapack_ A
+--------------+-------------+-----+
|  1        0 0|5   8      _2|3 2 3|
|0.4        1 0|0 3.8     9.8|     |
|0.2 0.631579 1|0   0 _2.78947|     |
+--------------+-------------+-----+
     L               U
```

-

$$LU = L \times U$$

```
   (>{.a) +/ . * >1{a
5 8 _2
2 7  9
1 4  3
```

7 QR 分解

QR 分解とは行列 *A* を直交行列 *Q* と右上三角行列 *R* の積として表すことである。シュミットの直交化を用いる方法やハウスホルダー行列を用いる方法がある。

$$A = QR$$

$$Q^T A = R$$

1. スクリプトのロード

```
load 'math/lapack'
 load 'math/lapack/geqrf'
```

2. 例（非対称行列）

```
   ] Q=: 5 _1 0,2 3 1,: 1 3 _1
5 _1  0
2  3  1
1  3 _1
```

3. 実行

```
   geqrf_jlapack_ Q
+-----------------------------+----------------------------+
|0.912871 _0.387842 _0.127458|5.47723 0.730297   0.182574|
|0.365148   0.63606  0.679775|      0  4.29729 _0.0310273|
|0.182574  0.667088 _0.722261|      0        0    1.40204|
+-----------------------------+----------------------------+
          Q                          R
```

4. *J* の *Foreign* にある *QR* 分解関数との比較
左の *Q* はシュミットの直交化になっている

```
   128!:0 Q
+--------------------------+-------------------------+
|0.912871 _0.387842 _0.127458|5.47723 0.730297   0.182574|
|0.365148   0.63606  0.679775|      0  4.29729 _0.0310273|
|0.182574  0.667088 _0.722261|      0        0    1.40204|
+--------------------------+-------------------------+
```

付録 A　LAPACK オブジェクト

1. オーバーラップ。　*J* はオーバーラップした関数や変数は警告なしに、後入れを優先する。*1* 枚ごとのファイルは自己責任でオーバーラップを管理し、多くのファイルをロードする場合は、目の届く範囲でラベルを張ったタンスの引き出しにまとめて収納する。この引き出しをオブジェクトとして扱う。

   LAPACK では `coclass 'jlapack'` で引き出しを決め、この引き出しにある関数や変数を使うには関数名の後ろに `xxx_jlapack_` を付ける。

   これ以外にオブジェクトに関するこ煩い規則はない。

2. *LAPACK* オブジェクト
   - 個別の計算関数には `_jlapack_` を付ける [*4]

     ```
     ] a0=. geev_jlapack_ A0
     ```
   - 面倒なら再定義する

     ```
     geev=: geev_jlapack_
     ```
   - チュートリアル *J8x* の *Help* →*Studio* → *Lab* → *LAPACK* にある。

     コマ送りは　*Ctrl*　*J* キーを同時に押す
   - 他にも線形計算用の *J* のスクリプトが沢山書かれているが、チュートリアルに入っていない。

     `,. routines_jlapack_ ''` で一覧を打ち出し、`addons/path/lapack` に含まれる *J* のスクリプトを読み解くしかないようだ。

付録 B　ミゼランス

J から *LAPACK* を使うには *J* のスクリプトで *LAPACK* との連携が要る。*J* のチュートリアルに入っていない *LAPACK* の線形関数がいくつか使用できる。

解説済み　.

1. *lapack.ijs*
2. *geev.ijs*

[*4] J のロケール機能を用いている

3. *gesvd.ijs*
4. *geqrf.ijs*
5. *getrf.ijs*
6. *heev.ijs NB.* エルミート行列の固有値と固有ベクトル
7. *potrf.ijs*

チュートリアルのないもの　スクリプトに用法がかかれている

1. *gebal.ijs NB. balances a general square matrix*
2. *gees.ijs NB. Schur* 分解
3. *gehrd.ijs* *NB.* 上 *Hessenberg* 行列に分解
4. *gelqf.ijs* *NB. LQ* 分解
5. *gels.ijs NB.* 行列式の不整合な行列の処理
6. *gelss.ijs* *NB. SVD* 分解を経由した最少自乗法の計算
7. *geqlf.ijs* *NB. QL* 分解
8. *gerqf.ijs* *NB. RQ* 分解
9. *gesv.ijs* *NB. A*X=B* の解
10. *gesvx.ijs* *NB.* 同上
11. *lartg.ijs* *NB.* ２つのベクトルの回転
12. *trtrs.ijs* *NB.* ３角行列から *A*X=B* を求める

LAPACK は最終的には *FORTRUN* で２バイト配列で処理される。関数の頭に *d* とあるのは実数、*z* とあるのは複素数である。次の関数は上の *gxxx.ijs* の頭に夫々 *d,z* を付けて名前のみ再定義したものである

1. *dgees.ijs*
2. *dgeev.ijs*
3. *dgeqrf.ijs*
4. *dgerqf.ijs*
5. *dgesv.ijs*
6. *dgesvd.ijs*
7. *dgetrf.ijs*
8. *dpotrf.ijs*
9. *dtrtrs.ijs*

1. *zgees.ijs*
2. *zgeev.ijs*
3. *zgeqrf.ijs*
4. *zgerqf.ijs*
5. *zgesv.ijs*
6. *zpotrf.ijs*
7. *ztrtrs.ijs*

References

Brain Bradie [A friendly Introduction to Numerical Analysis] Pearson Education 2006
木村　英紀 *[線形代数-数理科学の基礎]*　東京大学出版会　*2003*
薩摩順吉　四ツ谷昌二「キーポイント線形代数」　岩波書店　*1992*
町田 ·　駒崎　· 松浦「マトリクスの固有値と対角化」東海大学出版会　*1990*