

計算物理学 2 第 8 回：行列計算ライブラリ

ver. 2018/6/13

連立一次方程式や固有値問題などの線形計算は物理学以外の分野でも現れる共通の問題であり、自ら計算コードを作らなくても使用できるライブラリがあります。これらは多くのユーザによって使われており、自分でコードを構築するよりも早く正確に計算することができます。対角化などの行列計算は行列の次元が大きくなると計算時間が増大するため、できる限り行列計算ライブラリを使用することをおすすめします。この資料では LAPACK ライブラリの使用方法について説明します。

1 LAPACK

LAPACK(Linear Algebra PACKage, <http://www.netlib.org/lapack/>) は線形計算の Fortran パッケージです。C 言語から呼び出して使うことも可能です。LAPACK には様々なサブルーチンが含まれていますが、これらはドライバルーチン、計算ルーチン、補助ルーチンに分かれており、通常はドライバルーチンの中から目的にあったものを探して使います。ドライバルーチンには

- 線形方程式問題
- 線形最小二乗法問題・一般線形最小二乗法問題
- 固有値問題、特異値問題・一般化固有値問題・一般化特異値問題

などの問題に対するものが含まれています。

LAPACK のサブルーチンの名前は 6-7 文字程度で、用いる行列の種類、解きたい問題によってサブルーチンの名前が異なります。初めの一文字は変数型を表しており、

- S: 単精度実数
- D: 倍精度実数
- C: 単精度複素数
- Z: 倍精度複素数

のいずれかになっています。次の二文字は行列の種類を示しており、

- SY: 対称行列
- HE: エルミート行列
- GT: 三重対角行列
- ST: 対称三重対角行列
- GE: 一般行列

などがあります。

残りの文字は計算の種類を示しており、例えば線形方程式問題は SV で表され、倍精度実一般行列の線形方程式問題のルーチンは DGESV となります。倍精度複素エルミート行列の線形方程式問題を解く場合は ZHESV などとなります。

その他、固有値問題は EV となり、倍精度実対称行列の固有値問題は DSYEV、倍精度一般行列の固有値問題は DGEEV です。

LAPACK で提供されているルーチンのリストは例えば <http://www.netlib.org/lapack/explore-html/> にあります。これらの中からどのルーチンを使うべきかをまず決定します。

2 LAPACK のインストール

サテライト室のシステムには LAPACK ライブラリはインストールされており、gfortran で使うことができます。自分のコンピュータにインストールすることももちろんできますがここでは省略します。

3 LAPACK を使ったコーディング

LAPACK のルーチンはサブルーチン副プログラムとして呼び出すことができますので CALL 文を一文書くだけでライブラリを使うことができます。引数で計算に用いる行列の次元、行列、計算に一時的に必要な配列などを形式通り渡す必要がありますので引数の形式を合わせる必要があります。使いたいルーチン名で検索をしても見つけることができますし、LAPACK のルーチンの引数 (およびソースコード) のリストは <http://www.netlib.org/lapack/explore-html/> にもあります。

ここでは比較的良好に用いられる倍精度実行列の線形方程式ルーチン DGESV、倍精度実対称行列の固有値問題ルーチン DSYEV、倍精度実一般行列の固有値問題ルーチン DGEEV について説明します。LAPACK のルーチンには正常終了したかどうかを返す変数がありますが、いずれの場合も正常終了しなかった場合は予定されている値が計算されていないため、サブルーチンを呼んだ後はエラー処理をするようにしてください。

3.1 DGESV

倍精度実一般行列 A による連立一次方程式

$$AX = B \tag{1}$$

を解くルーチンです。ここで X や B は一般に行列です。使うときは CALL 文で呼び出します。

CALL DGESV (N, NRHS, A, LDA, IPIV, B, LDB, INFO)

ここで引数は

```
integer N
integer NRHS
double precision, dimension( LDA, N ) A
integer LDA
integer, dimension( N ) IPIV
double precision, dimension( LDB, NRHS ) B
integer LDB
integer INFO
```

の形式です。

このサブルーチンの入力変数は

- N は配列 A の 2 つ目の次元
- NRHS は配列 B の 2 つ目の次元 (解きたい N 元連立一次方程式の数)。
- A は LDA×N 次元の 2 次元配列 A(連立一次方程式の係数行列)。
- LDA は行列 A の 1 つ目の次元 (通常は N とします)。
- B は LDB×NRHS 次元の行列 B、LDB は行列 B の 1 つ目の次元 (通常は N とします)。

このサブルーチンの出力変数は

- 2 次元配列 A には行列 A を代入した状態でサブルーチンに値を渡しますが、実行後には値が書き換えられ、行列 A を LU 分解した結果が代入されます。L 行列と U 行列はそれぞれ下三角行列、上三角行列で、対角要素を除いて一つの $N \times N$ 行列に値を入れることができますが、L の対角要素は 1 であるため、U の対角要素が A として返されます。
- 2 次元配列 B には行列 B の値を入力しましたが、出力では方程式の解の X の値が返されます。
- IPIV は次元 N の整数型の一次元配列で、ピボットと呼ばれます。連立一次方程式を解く際に数値誤差を小さくするためにサブルーチンの中で行の順番を入れ替えることがありますが、その時の入れ替えた順番を保存するための配列です。
- INFO にはプログラムの実行結果が保存されます。正常終了した場合は 0 が入っています。負の数-i が返された場合は、i 行目で不正な値があったことを示します。また正の数 i が返された場合、LU 分解の U 行列の U_{ii} がゼロとなり (つまり $\det A = 0$ となり) 連立一次方程式の解が求まらなかったことを示します。

演習問題 30、31 で DGESV を使ったサンプルプログラムがありますので参照してください。

3.2 DSYEV

DSYEV は実対称行列 A の固有値問題

$$AU = UD \quad (2)$$

を解くサブルーチンです。呼び出すときは

CALL DSYEV (JOBZ,UPLD,N,A,LDA,W,WORK,LWORK,INFO)

の引数を指定します。引数の型は

```
character  JOBZ
character  UPLD
integer    N
double precision, dimension( LDA, N )  A
integer    LDA
double precision, dimension( N )  W
double precision, dimension( LWORK )  WORK
integer    LWORK
integer    INFO
```

入力時に値を設定する引数は

- JOBZ が 'N' の場合は固有値のみを計算、'V' の場合は固有値と固有ベクトルを計算します。
- UPLO: A は対称行列ですので 'U' の場合は A の上三角の部分に行列要素が入っているとして計算します。'L' の場合は下三角の部分の行列要素を使って計算します。
- N: 配列 A の次元
- A: 対角化したい行列 A で次元は $LDA \times N$ です。UPLO でどちらを指定するかによって計算時に参照される要素が変わります。
- LDA: 配列 A の 1 つ目の次元ですが通常 N とします。
- WORK: 作業用の倍精度実数型の LWORK 成分ある配列です。
- LWORK: WORK の要素数で、 $\max(1, 3N - 1)$ よりも大きく取るようにします。より大きい方が計算の効率があがります。

出力は

- A: 固有ベクトルを計算するオプションを指定していた場合はルーチン実行後の A には固有ベクトルが格納されます (U 行列が入ります)。固有ベクトルを計算しない場合ももとの入力時の A の値は破壊されます。
- W: 倍精度実数の N 要素の一次元配列で、小さいものから順に固有値が代入されます。実対称行列の固有値は実数です。
- INFO: 正常終了したら 0 が入ります。ゼロでない値が返された場合はマニュアルを参照してください。

演習問題 32 に DSYEV を使ったサンプルプログラムがあります。

3.3 DGEEV

DGEEV では実一般行列の固有値問題を解きます。DSYEV と類似していますが、対称行列に限らないため、左、右固有ベクトルを計算するオプションや、固有値は一般的に複素数となり、引数の構造が少し異なります。

CALL DGEEV (JOBVL, JOBVR, N, A, LDA, WR, WI, VL, LDVL, VR, LDVR, WORK, LWORK, INFO)

引数の形式は

```
character  JOBVL
character  JOBVR
integer    N
double precision, dimension( LDA, N )  A
integer    LDA
double precision, dimension( * )  WR
double precision, dimension( * )  WI
double precision, dimension( LDVL, * )  VL
integer    LDVL
```

```
double precision, dimension( LDVR, * )  VR
integer  LDVR
double precision, dimension( * )  WORK
integer  LWORK
integer  INFO
```

入力時に値を設定する変数は

- JOBVL 左固有ベクトルを計算する ('V') かしない ('N') かを与えます。
- JOBVR 右固有ベクトルを計算する ('V') かしない ('N') かを与えます。
- N A 行列の 2 つ目の次元です。
- A 対角化する LDA \times N 行列を入力します。ルーチン実行後は値は破壊されます。
- LDA A 行列の 1 つ目の次元で通常 N に設定します。
- WORK 作業用の要素数 LWORK の一次元配列です。
- LWORK WORK の次元。固有ベクトルを計算する場合は 4N よりも大きな値とします。

出力時に値が返ってくる変数は

- WR 一次元倍精度実数配列 (要素数 N)。固有値の実部が入ります。
- WI 一次元倍精度実数配列 (要素数 N)。固有値の虚部が入ります。
- VL LDVL \times N の二次元配列。左固有ベクトルを計算するオプションを選んでいたら、左固有ベクトルが入ります。
- LDVL VL の次元。通常 N に設定します。
- VR LDVR \times N の二次元配列。右固有ベクトルを計算するオプションを選んでいたら、右固有ベクトルが入ります。
- LDVR VR の次元。通常 N に設定します。
- INFO 正常終了した場合はゼロが代入されます。そうでない場合はマニュアルを参照してください。

複素数固有値が出る場合は固有ベクトルへの値の格納の仕方が変わります。詳しくはマニュアルを参照してください。演習問題 33 に DGEEV を使ったサンプルプログラムがあります。

3.4 C 言語での LAPACK を用いたコーディング

LAPACK は Fortran で書かれていますので C 言語からは Fortran のサブルーチンを呼び出すという方法で用います。例えば DGEV を使う場合は C 言語では最後にアンダースコアをつけた関数 dgevs_() を呼び出します。引数の形式は基本的に Fortran と同じですが、違いは多次元配列は一次元配列として定義し、すべてポインタ渡しをします。これは Fortran のサブルーチンは参照渡しをしているためです。dgevs の場合は

```
dgevs_(&n, &nrhs, a, &lda, ipiv, b, &ldb, &info);
```

ここで a は要素数 lda \times n の一次元配列、b は要素数 ldb \times nrhs の一次元配列です。(配列の先頭要素のアドレスはそれぞれ a, b で与えられますが変数のアドレスは変数名に&をつけて与えられます)

4 コンパイル

4.1 Fortran

Fortran では LAPACK はサブルーチンとして呼び出されます。このままコンパイルすると、このサブルーチンがプログラムの中で定義されていないため、エラーがでます。例えば演習問題 30 のコードをコンパイルすると

```
$ gfortran exercise30.f90 -o exercise30.exe
/tmp/ccsUdakv.o: 関数 'MAIN_' 内:
exercise30.f90:(.text+0x410): 'dgesv_' に対する定義されていない参照です
collect2: error: ld returned 1 exit status
```

といった、中で呼び出されている”DGESV”が見つからないというエラーがでます。LAPACK を使うためにはコンパイル時にライブラリをリンクする必要があります。これはコンパイルオプション `-llapack -lblas` を使います。`-l` ライブラリ名 で指定されたライブラリを見に行きますが、LAPACK の中では BLAS が使われていますので BLAS ライブラリも同時にリンクする必要があります。`-l` とライブラリ名の間にスペースは要りません。

```
$ gfortran exercise30.f90 -o exercise30.exe -llapack -lblas
```

4.2 C 言語

C 言語でも通常のコンパイルを行うと

```
$ gcc exercise30.c -o exercise30.exe
/tmp/cc0qRgMv.o: 関数 'main' 内:
exercise30.c:(.text+0x30e): 'dgesv_' に対する定義されていない参照です
collect2: error: ld returned 1 exit status
```

とエラーがでます。`-llapack -lblas` オプションを用いて

```
$ gcc exercise30.c -o exercise30.exe -llapack -lblas
```

のようにコンパイルします。

5 演習問題

10×10 行列

$$A = \begin{pmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 9 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 8 & 9 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 \\ 7 & 8 & 9 & 10 & 9 & 8 & 7 & 6 & 5 & 4 \\ 6 & 7 & 8 & 9 & 10 & 9 & 8 & 7 & 6 & 5 \\ 5 & 6 & 7 & 8 & 9 & 10 & 9 & 8 & 7 & 6 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 & 8 & 7 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix} \quad (3)$$

を考える。

(30) 行列 A と列ベクトル

$$\mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ -6 \\ -7 \\ -8 \\ -9 \\ -10 \end{pmatrix} \quad (4)$$

からなる連立一次方程式

$$A\mathbf{x} = \mathbf{b} \quad (5)$$

を LAPACK の実線形方程式のルーチン DGESV を用いて計算せよ。

- (31) 行列 A の行列式 $\det A$ および逆行列 A^{-1} を DGESV を用いて求めよ。
- (32) 行列 A を実対称行列の固有値問題のルーチン DSYEV を用いて対角化し、固有値と固有ベクトルを求めよ。
- (33) 行列 A を実一般行列の固有値問題のルーチン DGEEV を用いて対角化し、固有値と固有ベクトルを求めよ。