

線形代数パッケージ LAPACK について

浅岡 香枝* 平野 彰雄**

1 はじめに

線形代数パッケージ LAPACK (Linear Algebra PACKage) は、EISPACK (固有値問題) ならびに LINPACK (連立方程式) パッケージの後継として米国アルゴンヌ国立研究所を中心に欧米の研究者が数年に渡って共同開発された Fortran サブルーチンライブラリです。

LAPACK は豊富なドキュメントとともにソースプログラムがインターネットに公開されており、個人でも入手しワークステーションやパーソナルコンピュータなどに移植して利用することができます。

近年、計算機が高性能化するなかで、ワークステーションやパーソナルコンピュータなどの環境でプログラミングおよびデバックを行い、大規模な計算には本センターのスーパーコンピュータで利用も多くなっています。このようなプログラミング開発環境で使用する数値計算ライブラリとしては、ソースプログラムが公開され色々なシステムに移植されている LAPACK が大変有効であると考え、センターのスーパーコンピュータ用にベクトル化チューニングを施し、LAPACK を移植し公開しました。

ここでは、センターでの LAPACK の使い方と LAPACK の構成と機能について解説します。

2 LAPACK の使い方

2.1 移植した LAPACK

センターで移植した LAPACK は、1994 年 10 月に公開された第二版です。

移植は会話型ベクトル計算機 (abacus) の Fortran90 コンパイラで行い、

- 会話型ベクトル計算機 (abacus)
- ベクトル並列計算機 (VPP500) の 1PE

で使えるようにベクトルチューニングしています。

なお、移植した LAPACK のソースプログラムおよびオリジナルとの差分ファイルなどを移植作業の履歴と共に、公開しています。

2.2 コンパイルとライブラリの指定

LAPACK パッケージは、線形代数ライブラリ LAPACK と基本線形代数サブルーチン BLAS から構成されています。BLAS は線形代数演算の核となるルーチンであり、LAPACK を使用する場合には必須です。

コンパイルは `f90` コマンドを使って次のような指定になります。

```
abacus% f90 sample.f -llapack -lblas
```

ここで、`sample.f` はソースファイル名で、あとに LAPACK (`-llapack`), BLAS (`-lblas`) の順にライブラリを指定します。

BLAS ライブラリを単独で利用することも可能で、この場合には、次のように指定します。

```
abacus% f90 sample.f -lblas
```

2.3 オンラインマニュアル

abacus では LAPACK, BLAS のサブルーチンを `man` コマンドで調べることができます。

キーワードでどのようなルーチンがあるかを調べるには次のように指定します。

```
abacus% man -k equations | less
```

また、この中の一般行列の連立一次方程式を解く倍精度ルーチン DGESV の呼出し形式および引数の意味を知るには次のように指定します。

```
abacus% man dgesv
```

* あさおか かえ, ** ひらの あきお (京都大学大型計算機センター)

3 LAPACK の基礎

3.1 LAPACK が扱う問題

LAPACK が扱う線形代数問題は、次のものです。

- 連立一次方程式
- 線形の最小二乗問題
- 固有値問題
- 特異値問題

3.2 ルーチンの構成

ルーチンには単精度、倍精度実数および複素数、倍精度複素数が用意されてます。また、各ルーチンは、その機能からドライバルーチン、計算ルーチン、補助ルーチンに分られます。

ドライバルーチンとは、連立一次方程式を解く、固有値を求めるなど、それぞれで問題を解くことができるルーチンです。

計算ルーチンとは、ドライバルーチンの下位のルーチンで、行列の分解や変換、分解した行列を使って解を求めるなど、部分の計算を行うもので、これらを組み合わせることで、ドライバルーチンが提供しない問題を扱うことが可能になります。

補助ルーチンとは、行列のスケーリングなどより下位の計算を行うルーチンです。

なお、ベクトルと行列の乗算などの基本演算の多くは BLAS ルーチンが使われています。

3.3 シンプルとエキスパート

さらに、ドライバルーチンには、連立一次方程式の解の反復改良や固有値を区間指定で求めたりするといった高機能なルーチンも提供されています。これらは、一般の解を求めるルーチン (シンプル) と対比し (エキスパート) と呼ばれています。

エキスパートドライバは、高機能であるがゆえに CPU 時間が増え、メモリも多く必要となりますので、注意が必要です。

3.4 ルーチン名

LAPACK のルーチン名は FORTRAN77 の 6 文字という制約の中で、次のようなルールで作られています。

1) 先頭の 1 文字 ルーチンが扱うデータの型を (S) 実数、(D) 倍精度実数、(C) 複素数、(Z) 倍精度複素数で示します。

2) 次の 2 文字 ルーチンが扱う行列の型を示しています。例えば、(GE) 一般行列、(GB) 一般帯行列などを意味します。

3) 残りの文字 (2 あるいは 3 文字) ルーチンが扱う演算の型を示しています。例えば、(SV) 連立一次方程式、(EV) 対称固有値などを意味します。

3.5 ドライバルーチンの一覧

LAPACK のドライバルーチンのルーチン名と機能の一覧を表 1、表 2、表 3、表 4 に示します。なお、ルーチン名の最後に X がついているものがエキスパートドライバです。

4 LAPACK の情報

LAPACK を使う上で役立つ情報源には、次のものがあります。

1) <http://www.netlib.org/lapack/>

ここが LAPACK のホームです。移植のためのプログラムや利用の手引き、アルゴリズムなどの解説書である LAPACK Working Note が入手できます。

2) <http://phase.etl.go.jp/netlib/lapack/netlib> の国内のミラーサイトです。

3) 利用の手引きの訳本

「行列計算パッケージ LAPACK 利用の手引」
小国力 訳, 丸善, ISBN 4-621-04076-6

5 おわりに

LAPACK の使い方と LAPACK 構成について簡単に解説しました。

センターで移植した LAPACK のソースプログラムなどは、

<http://www.kudpc.kyoto-u.ac.jp/HPC/>

で公開しています。ご意見、ご質問などは、

hpc-wg@kudpc.kyoto-u.ac.jp

までお寄せ下さい。

表 1. 連立一次方程式を解くルーチン

実数	複素数	説 明
DGESV	ZGESV	一般行列の連立一次方程式 $AX = B$ を解く
DGESVX	ZGESVX	一般行列の連立一次方程式 $AX = B$ 、 $A^T X = B$ 、 $A^H X = B$ を解き、条件数と解の誤差限界を評価する
DGBSV	ZGBSV	一般帯行列の連立一次方程式 $AX = B$ を解く
DGBSVX	ZGBSVX	一般帯行列の連立一次方程式 $AX = B$ 、 $A^T X = B$ 、 $A^H X = B$ を解き、条件数と解の誤差限界を評価する
DGTSV	ZGTSV	一般三重対角行列の連立一次方程式 $AX = B$ を解く
DGTSVX	ZGTSVX	一般三重対角行列の連立一次方程式 $AX = B$ 、 $A^T X = B$ 、 $A^H X = B$ を解き、条件数と解の誤差限界を評価する
DPOSV	ZPOSV	対称 / エルミート正定値行列の連立一次方程式 $AX = B$ を解く
DPOSVX	ZPOSVX	対称 / エルミート正定値行列の連立一次方程式 $AX = B$ を解き、条件数と解の誤差限界を評価する
DPPSV	ZPPSV	圧縮型対称 / エルミート正定値行列をもつ連立一次方程式 $AX = B$ を解く
DPPSVX	ZPPSVX	圧縮型対称 / エルミート正定値行列をもつ連立一次方程式 $AX = B$ を解き、条件数と解の誤差限界を評価する
DPBSV	ZPBSV	対称 / エルミート正定値帯行列をもつ連立一次方程式 $AX = B$ を解く
DPBSVX	ZPBSVX	対称 / エルミート正定値帯行列をもつ連立一次方程式 $AX = B$ を解き、条件数と解の誤差限界を評価する
DPTSV	ZPTSV	対称 / エルミート正定値三重対角行列をもつ連立一次方程式 $AX = B$ を解く
DPTSVX	ZPTSVX	対称 / エルミート正定値三重対角行列をもつ連立一次方程式 $AX = B$ を解き、条件数と解の誤差限界を評価する
DSYSV	ZSYSV ZHESV	実対称 / 複素対称 / 複素エルミート不定値行列をもつ連立一次方程式 $AX = B$ を解く
DSYSVX	ZSYSVX ZHESVX	実対称 / 複素対称 / 複素エルミート不定値行列をもつ連立一次方程式 $AX = B$ を解き、条件数と解の誤差限界を評価する
DSPSV	ZSPSV ZHPSV	圧縮型実対称 / 複素対称 / 複素エルミート不定値行列をもつ連立一次方程式 $AX = B$ を解く
DSPSVX	ZSPSVX ZHPSVX	圧縮型実対称 / 複素対称 / 複素エルミート不定値行列をもつ連立一次方程式 $AX = B$ を解き、条件数と解の誤差限界を評価する

表 2. 最小二乗問題を解くルーチン

実数	複素数	説 明
DGELS	ZGELS	A の QR 分解、 LQ 分解を用いて、過剰定義の連立一次方程式 $AX = B$ 、 $A^T X = B$ 、 $A^H X = B$ の最小二乗解を計算するか、過小定義系の最小ノルム解を計算する。ここで、 A は充足階数の一般長方形行列である。
DGELSS	ZGELSS	A の特異値分解を用いて、過剰・過少定義の連立一次方程式 $AX = B$ の最小ノルム最小二乗解を計算する
DGELSX	ZGELSX	A の完全直交分解を用いて、過剰・過少定義の連立一次方程式 $AX = B$ の最小ノルム解を計算する
DGGLSE	ZGGLSE	A の一般化 RQ 分解を用いて線形等式制約最小二乗問題を解く
DGGGLM	ZGGLSE	A の一般化 QR 分解を用いて一般ガウス・マルコフ線形モデル問題を解く

表 3. 固有値問題を解くルーチン

実数	複素数	説 明
DSYEV	ZHEEV	対称 / エルミート行列の全ての固有値と固有ベクトルを計算する
DSYEV	ZHEEVD	対称 / エルミート行列の全ての固有値と固有ベクトルを計算する。固有ベクトルは divide and conquer 法を用いて計算をする
DSYEVX	ZHEEVX	対称 / エルミート行列の指定した固有値と固有ベクトルを計算する。
DSPEV	ZHPEV	圧縮型対称 / エルミート行列の全ての固有値と固有ベクトルを計算する
DSPEVD	ZHPEVD	圧縮型対称 / エルミート行列の全ての固有値と固有ベクトルを計算する。固有ベクトルは divide and conquer 法を用いて計算をする
DSPEVX	ZHPEVX	圧縮型対称 / エルミート行列の指定した固有値と固有ベクトルを計算する。
DSBEV	ZHBEV	対称 / エルミート帯行列の全ての固有値と固有ベクトルを計算する
DSBEVD	ZHBEVD	対称 / エルミート帯行列の全ての固有値と固有ベクトルを計算する。固有ベクトルは divide and conquer 法を用いて計算をする
DSBEVX	ZHBEVX	対称 / エルミート帯行列の指定した固有値と固有ベクトルを計算する。
DSTEV		実対称三重対角行列の全ての固有値と固有ベクトルを計算する
DSTEV	DSTEVD	実対称三重対角行列の全ての固有値と固有ベクトルを計算する。固有ベクトルは divide and conquer 法を用いて計算をする
DSTEVX		実対称三重対角行列の指定した固有値と固有ベクトルを計算する。
DGEES	ZGEES	一般行列の固有値と Schur 分解を計算し、指定した固有値が Schur 形の左上にくるように並べる
DGEESX	ZGEESX	一般行列の固有値と Schur 分解を計算し、指定した固有値が Schur 形の左上にくるように並べ、指定した固有値の平均および対応する右不変部分空間に対する条件数の逆数を計算する
DGEEV	ZGEEV	一般行列の固有値と左右の固有ベクトルを計算する
DGEEVX	ZGEEVX	あらかじめ行列を均衡化して一般行列の固有値と左右の固有ベクトルを計算し、固有値と右固有ベクトルに対して条件数の逆数を計算する
DSYGV	ZHEGV	対称 / エルミート定値行列の一般化固有値問題 $Ax = \lambda Bx, ABx = \lambda x, BAx = \lambda x$ の全ての固有値と固有ベクトルを計算する
DSPGV	ZHPGV	圧縮型対称 / エルミート定値行列の一般化固有値問題 $Ax = \lambda Bx, ABx = \lambda x, BAx = \lambda x$ の全ての固有値と固有ベクトルを計算する。
DSBGV	ZHBGV	対称 / エルミート定値帯行列の一般化固有値問題 $Ax = \lambda Bx$ の全ての固有値と固有ベクトルを計算する。
DGEGS	ZGEGS	一对の非対称行列に対して一般化固有値、Schur 形、左右の Schur ベクトルを計算する
DGEGV	ZGEGV	一对の非対称行列に対して一般化固有値、左右の一般化固有ベクトルを計算する

表 4. 特異値問題を解くルーチン

実数	複素数	説 明
DGESVD	ZGESVD	一般長方形行列の特異値分解を計算する
DGGSVD	ZGGSVD	一对の一般長方形行列の一般化特異値分解を計算する