

# 汎用最小二乗推定 FORTRAN ライブラリ (liblsqf) 説明書

近藤哲朗

2005 年 6 月 19 日版

# 目次

1	概要	1
2	プログラム作成法	1
2.1	間接法	1
2.2	直接法	1
3	最小二乗推定ライブラリ (liblsqf) の共通配列および変数の意味	3
4	liblsqf ルーチン群の説明 (下線はリターンパラメータ)	3
4.1	LSQINIT	3
4.2	LSQGETMAX( <u>MAXOBS</u> , <u>MAXPARA</u> )	3
4.3	LSQSETSIZE( <u>NOBS</u> , <u>NPARA</u> )	4
4.4	LSQSETOBS( <u>Y</u> , <u>YERR</u> , <u>N</u> )	4
4.5	LSQSETJACOB( <u>A</u> , <u>N</u> , <u>M</u> , <u>IERR</u> )	4
4.6	DOLSQ( <u>IERR</u> )	4
4.7	LSQGETKAI( <u>X</u> , <u>XERR</u> , <u>M</u> )	4
4.8	LSQGETCOV( <u>V</u> , <u>M1</u> , <u>M2</u> )	4
4.9	LSQGETCOR( <u>C</u> , <u>M1</u> , <u>M2</u> )	4
4.10	LSQGETSTD( <u>RSUM2</u> , <u>RSIGMA</u> , <u>RCHI2</u> , <u>NFREE</u> , <u>NDATA</u> , <u>MPARA</u> )	4
4.11	LSQGETOMC( <u>R</u> , <u>N</u> )	5
4.12	LSQSETFLAG( <u>KEY</u> , <u>K</u> , <u>AFLAG</u> )	5
4.13	LSQSETFLAG( <u>KEY</u> , <u>K</u> , <u>IFLAG</u> )	5
4.14	LSQDEBUG	5

# 1 概要

1991年にHP1000/A900のFORTRANで作成したサブルーチン群を汎用のFORTRANで記述。最小二乗解の計算は特異値分解法によっている。ランク落ちの起きたパラメータに対しても最小二乗最小ノルム解(答えの善し悪しは別としてももっともらしい解)が得られる。また次元あり重みを採用しているので異なる次元(単位)の観測量も同時に扱える。なお、特異値分解プログラムの出典は「計算機のための数値計算法」G.E.Forsythe/M.A.Malcolm/C.B.Moler 著、森正武訳、科学技術出版社(1978)。

取り扱える最大パラメータ数は100、取り扱える最大データ数は2500であるが、LSQ\_F\_INCLUDE.TXTの配列宣言部を書き換えることにより、変更可能である。

## 2 プログラム作成法

### 2.1 間接法

liblsqf ルーチン群が内部で使用している共通配列への必要データのセットおよびデータのゲットを liblsqf ルーチン群を用いて行う。メインプログラムで liblsqf ルーチン群で使用する共通配列を意識する必要がなく、ユーザーにとってはプログラムの作成が容易となる。

(プログラム例)

```
PROGRAM main
....
....
CALL LSQINIT                .... liblsqf 群の初期化
DO N=1,NOBS
....
CALL LSQSETOBS(Y,YERR,N)   .... 観測データ、誤差のセット
DO M=1,NPARA
....
CALL LSQSETJACOB(ANM,N,M,IERR) .... ヤコビアン of the セット
END DO
END DO
....
CALL LSQSETSIZE(NOBS,NPARA) .... 実際のヤコビアン行列のサイズを教える
CALL DOLSQ(IERR)           .... 最小二乗推定の実行
CALL LSQGETKAI(X,XERR,m)   .... 推定値(アジャスト値)を得る
                             m はパラメータ番号
XA(m)=XA(m)+X              .... 真の解はアプリアリ値 XA にアジャスト値 X を加えたもの
....
```

### 2.2 直接法

liblsqf ルーチン群が内部で使用している共通配列に直接データをセットしたり、共通配列から直接解を得る方法。

(プログラム例)

```
PROGRAM main
```

```

    IMPLICIT REAL*8(D)      .... 必ず必要
$INCLUDE LSQ_F_INCLUDE.TXT  .... 必ず必要
    ....
    ....
M_PMAX=100                  .... この通りにすること(最大可能パラメータ数)
M_OMAX=2500                 .... この通りにすること(最大可能データ数)
    ....
DO N=1,NOBS
    ....
    R_Y(N)= ...              .... 観測データのセット
    R_YERR(N)= ...           .... 観測誤差のセット
    I_GOOD(N)= 1 or 0        .... 良好データフラグのセット(1:good)
    DO M=1,NPARA
        ....
        D_A(N,M)= ...        .... ヤコビアン(ヤコビ行列)のセット
    END DO
END DO
DO M=1,NPARA
    I_FLAG(M)= 1 or 0        .... 推定パラメータフラグのセット(1:推定)
END DO
N_OBS=NOBS                  .... 実際のデータ数をセット
N_PARA=NPARA                 .... 実際のパラメータ数をセット
CALL DOLSQ(IERR)             .... 最小二乗推定の実行
X=X_ADJ(m)                   .... 推定値(アジャスト値)を得る
                               mはパラメータ番号
XA(m)=XA(m)+X                .... 真の解はアプリアリ値 XA にアジャスト値 X を加えたもの
    ....

```

### 3 最小二乗推定ライブラリ (liblsqf) の共通配列および変数の意味

(ファイル LSQ\_F\_INCLUDE.TXT)

COMMON /LSQ/

定義

R_Y(2500)	観測データ (O - C データ) 収納配列
R_YERR(2500)	観測誤差データ収納配列
L_GOOD(2500)	観測データフラグ配列 1: 良好データ 0: 悪データ (推定に使用しない)
D_A(2500,100)	ヤコビアン行列
R_OMC(2500)	最小二乗推定後の O - C 残差収納配列
X_ADJ(100)	最小二乗推定パラメータ値 (アジャスト値) 収納配列
L_FLAG(100)	推定パラメータフラグ収納配列 1: 推定する 0: 推定しない 推定後-1 となったパラメータはランク落ちの発生したパラメータ
D_CORX(100,100)	得られたパラメータ間の相関係数行列収納配列
D_COVX(100,100)	得られたパラメータ間の共分散行列収納配列
R_SUM2	推定後の規格化残差二乗 $\chi^2$ 今の場合 $\chi^2 = S(\hat{x}) = \tilde{v}(\hat{x})\Sigma^{-1}v(\hat{x})$ ただし $v(\hat{x}) = y - f(\hat{x})$ : 残差 $\Sigma^{-1}$ : 推定パラメータの誤差行列 (D_COVX) の逆行列 (スケーリング前)
R_SIGMA	推定後の残差の標準偏差 ( $\sigma_*$ ) $\sigma_*^2 = S(\hat{x})/(n - m)$ ただし $n - m$ : 自由度
R_CHI2	規格化 $\chi^2 = \chi^2 / \text{自由度} (n - m) = \sigma_*^2$
N_FREE	推定後の自由度 (観測データ数 $n$ - 推定パラメータ数 $m$ )
M_PMAX	最大可能パラメータ数
M_OMAX	最大可能データ数
N_DATA	実際のデータ数 (bad データも含む)
N_OBS	Good データ数
N_PARA	推定パラメータ数
M_PARA	実際に推定されたパラメータ数 (ランク落ちパラメータも含む)
LDBK	処理デバッグ情報出力の制御 1: ON 0: OFF

### 4 liblsqf ルーチン群の説明 (下線はリターンパラメータ)

#### 4.1 LSQINIT

内部配列の初期化。観測フラグはすべて 0、パラメータフラグはすべて 1 に初期化される。このルーチンは必ず最初に呼んでおくこと。

#### 4.2 LSQGETMAX(MAXOBS,MAXPARA)

使用可能な最大データ数 (MAXOBS)、および最大パラメータ数 (MAXPARA) を得る。デフォルトではそれぞれ 2 5 0 0、1 0 0 がリターンされる。

### 4.3 LSQSETSIZE(NOBS,NPARA)

ヤコビアン行列のサイズ(実データ数および実パラメータ数)を最小二乗推定ルーチン DOLSQ に教える。DOLSQ に先立って必ずこの関数を呼ぶこと。

### 4.4 LSQSETOBS(Y,YERR,N)

N 番目の観測データ (Y) および誤差 (YERR) をセットする。Y は通常 O-C をセットする。このルーチンで良好なデータを示すフラグもセットされる。悪データ(推定に使用しないデータ)としたいときは LSQSETFLAG ルーチンでフラグを制御できる。

### 4.5 LSQSETJACOB(A,N,M,IERR)

N 番目の観測のパラメータ番号 M のヤコビアン(偏微分係数) A をセットする。N が最大可能データ多数を超えた場合、IERR=-1、M が最大可能パラメータ数を超えた場合、IERR=-2 がリターンされる。正常の場合は IERR=0 がリターンされる。

### 4.6 DOLSQ(IERR)

最小二乗推定の実施。正常終了の場合 IERR=0 がリターンされる。推定でエラーが発生した場合、IERR=-1 がリターンされる。

### 4.7 LSQGETKAI(X,XERR,M)

最小二乗推定で得たパラメータ番号 M の推定値(アジャスト値) X と 1  $\sigma$  誤差 XERR を得る。この誤差はすでに  $\sigma_*$  が乗じられている値(いわゆる scaled  $\sigma$ )。従ってそのまま推定誤差として用いることができる。

### 4.8 LSQGETCOV(V,M1,M2)

推定パラメータ M1 と M2 間の共分散値 V を得る。共分散値の推定値とするにはスケーリング定数  $\sigma_*^2$  (=R.SIGMA<sup>2</sup>) を乗じること。

### 4.9 LSQGETCOR(C,M1,M2)

推定パラメータ M1 と M2 間の相関係数 C を得る。

### 4.10 LSQGETSTD(RSUM2,RSIGMA,RCHI2, NFREE, NDATA, MPARA)

最小二乗推定後の各種統計量を得る。

- RSUM2 : 規格化残差二乗和 ( $\chi^2$ )  
 今の場合  $\chi^2 = S(\hat{x}) = \tilde{v}(\hat{x})\Sigma^{-1}v(\hat{x})$   
 ただし  $v(\hat{x}) = y - f(\hat{x})$  : 残差  
 $\Sigma^{-1}$  : 推定パラメータの誤差行列 (D\_COVX) の逆行列 (スケーリング前)
- RSIGMA : 推定後の規格化残差の標準偏差 ( $\sigma_*$ )  
 $\sigma_*^2 = S(\hat{x})/(n - m)$   
 ただし  $n - m$  : 自由度
- RCHI2 : 規格化  $\chi^2 = \chi^2 / \text{自由度} (n - m) = \sigma_*^2$
- NFREE : 自由度 (= データ数 - パラメータ数)
- NDATA : 実際に推定に用いられたデータ数
- MPARA : 実際に推定されたパラメータ数 (ランク落ちパラメータも含む)

#### 4.11 LSQGETOMC(R,N)

最小二乗推定後の観測番号 N の残差 R を得る

#### 4.12 LSQSETFLAG(KEY,K,AFLAG)

推定パラメータフラグのセット、リセットや観測データフラグのセット、リセットを行う。

パラメータフラグ指定の場合

KEY : 'PARAM' .. パラメータフラグ指定キー

K : パラメータ番号

AFLAG : 'YES' .. 推定する 'NO' .. 推定しない

観測データフラグ指定の場合

KEY : 'OBS' .. 観測データフラグ指定キー

K : 観測番号

AFLAG : 'GOOD' .. Good データとする 'BAD' .. Bad データとする

#### 4.13 LSQGETFLAG(KEY,K,IFLAG)

推定パラメータフラグおよび観測データフラグの値を得る。

パラメータフラグ指定の場合

KEY : 'PARAM' .. パラメータフラグ指定キー

K : パラメータ番号

AFLAG : 1 .. 推定する 0 .. 推定しない

-1 .. 推定パラメータとして指定したがランク落ち発生

観測データフラグ指定の場合

KEY : 'OBS' .. 観測データフラグ指定キー

K : 観測番号

AFLAG : 1 .. Good データ 0 .. Bad データ

#### 4.14 LSQDEBUG

最小二乗推定ルーチン内のデバッグ出力を指定する。LSQINIT を呼んだ後はデバッグ出力無しが指定されている。