

1.LAPACK 简介

1.1 概述

LAPACK API 支持两种形式：一是标准的 ANSI C；另一种是标准的 FORTRAN77。LAPACK 是开源的，官方网站是：<http://www.netlib.org/lapack/>。

每个 LAPACK 例程都有四个形式，具体描述如下：

精度	例程前缀
REAL 精度	S
REAL DOUBLE 精度	D
COMPLEX 单精度	C
COMPLEX 双精度	Z

下面例程是经过优化的。

DGETRF 对一般矩阵进行 LU 分解。

DGETRS 线性方程组求解。

DGETRI 用 LU 分解求解一般矩阵的逆矩阵。

DGEQRF 对一般矩阵进行 QR 分解。

DGELQF 对一般矩阵进行 LQ 分解。

DPOTRF 对对称正定矩阵进行 Cholesky 分解。

DPOTRS 对线性方程组（对称正定）求解。

1.2 函数的命名规则：

LAPACK 里的每个函数名已经说明了该函数的使用规则。所有函数都是以 XYYZZZ 的形式命名，对于某些函数，没有第六个字符，只是 XYYZZ 的形式。

第一个字母 X 代表以下的数据类型：

- S REAL，单精度实数
- D DOUBLE PRECISION，双精度实数
- C COMPLEX，单精度复数
- Z COMPLEX*16 或 DOUBLE COMPLEX

注：

在新版 LAPACK 中含有使用重复迭代法的函数 DSGESV 和 ZCDESV。

头 2 个字母表示使用的精度：

- DS 输入数据是 double 双精度，算法使用单精度
- ZC 输入数据是 complex*16，算法使用 complex 单精度复数

接下面两个字母 YY 代表数组的类型。

- BD bidiagonal，双对角矩阵
- DI diagonal，对角矩阵
- GB general band，一般带状矩阵
- GE general (i.e., unsymmetric, in some cases rectangular)，一般情形（即非对称，在有些情形下为矩形）
- GG general matrices, generalized problem (i.e., a pair of general matrices)，一般矩阵，广义问题（即一对一般矩阵）
- GT general tridiagonal，一般三对角矩阵
- HB (complex) Hermitian band，（复数）厄尔米特带状阵
- HE (complex) Hermitian，（复数）厄尔米特矩阵

HG	upper Hessenberg matrix, generalized problem (i.e a Hessenberg and a triangular matrix), 上海森伯格矩阵, 广义问题 (即一个海森伯格矩阵和一个三角矩阵)
HP	(complex) Hermitian, packed storage, (复数) 压缩储存的厄尔米特矩阵
HS	upper Hessenberg, 上海森伯格矩阵
OP	(real) orthogonal, packed storage, (实数) 压缩储存的正交阵
OR	(real) orthogonal, (实数) 正交阵
PB	symmetric or Hermitian positive definite band, 对称或厄尔米特正定带状矩阵
PO	symmetric or Hermitian positive definite, 对称或厄尔米特正定矩阵
PP	symmetric or Hermitian positive definite, packed storage, 压缩储存的对称或厄尔米特正定矩阵
PT	symmetric or Hermitian positive definite tridiagonal, 对称或厄尔米特正定三对角阵
SB	(real) symmetric band, (实数) 对称带状阵
SP	symmetric, packed storage, 压缩储存的对称阵
ST	(real) symmetric tridiagonal, (实数) 对称三对角阵
SY	symmetric, 对称阵
TB	triangular band, 三角形带状矩阵
TG	triangular matrices, generalized problem (i.e., a pair of triangular matrices), 三角形矩阵, 广义问题 (即一对三角形阵)
TP	triangular, packed storage, 压缩储存的三角形阵
TR	triangular (or in some cases quasi-triangular), 三角形阵 (在某些情形下为类三角形阵)
TZ	trapezoidal, 梯形阵
UN	(complex) unitary, (复数) 酉矩阵
UP	(complex) unitary, packed storage, (复数) 压缩储存的酉矩阵

最后三个字母 ZZZ 代表计算方法。比如, SGEBRD 是一个单精度函数, 用于把一个实数一般阵压缩为双对角阵 (a bidiagonal reduction, 即 BRD)。

2.LAPACK 详解

本部分以 **S** 系列例程为例, 本部分共有 **334** 个例程(**3.0** 版本)。其他的就大家参考本部分就可以了。此次先介绍 **FORTRAN** 版本的, **C** 版本的在以后做介绍, 或者参照本部分亦可以。

2.1 SBDSDC 子程序

原型:

SUBROUTINE SBDSDC(UPLO, COMPQ, N, D, E, U, LDU, VT, LDVT, Q, IQ,
\$ WORK, IWORK, INFO)

例程说明:

SBDSDC 是对 real 型的 $N \times N$ 阶对角矩阵 B ($B = U * S * VT$) 进行 SVD 计算。

【注: 奇异值分解 (singular value decomposition, SVD) 是一种正交矩阵分解法; SVD 是最可靠的分解法, 但是它比 QR 分解 (QR 分解法是将矩阵分解成一个正规正交矩阵与上三角形矩阵。) 法要花上近十倍的计算时间。】

参数说明:

UPLO 输入参数 (CHARACTER*1)

= 'U' : B 是上对角矩阵 (upper bidiagonal)

= 'L' : B 是下对角矩阵 (lower bidiagonal);

COMPQ 输入参数 (CHARACTER*1)

指定是否奇异向量按照如下方式计算:

= 'N' 只计算奇异值

= 'P' 计算奇异值和紧式 (compact form) 奇异向量

= 'I' 计算奇异值和奇异向量

N 输入参数 (INTEGER)

B 矩阵的秩。N>=0。

D 输入/输出参数, real 型数组, dimension (N)

输入时, 为 B 矩阵的 N 个对角元素

输出时, 如果 INFO = 0, 则为 B 矩阵的奇异值。

E 输入输出参数, real 型数组, dimension (N)

输入时, E 所包含的非对角矩阵元素,

输出时, E 被破坏。

U 输出参数, real 型数组, dimension (LDU,N)

如果 COMPQ = 'I', 输出时并且 INFO = 0, U 接受对角矩阵的左奇异向量, 对于其他的 COMPQ 数值, U 不被采用。

LDU 输入参数 (INTEGER)

决定 U 数组的尺寸, LDU>=1

若果想得到奇异向量, LDU>=max (1, N)。

VT 输出参数, real 型数组, dimension (LDVT,N)

如果 COMPQ = 'I', 输出时并且 INFO = 0, VT 接受对角矩阵的右奇异向量, 对于其他的 COMPQ 数值, VT 不被采用。

LDVT 输入参数, (INTEGER)

决定 VT 数组的尺寸, LDVT>=1

若果想得到奇异向量, LDVT>=max (1, N)。

Q 输出参数, real 型数组, dimension (LDQ)

如果 COMPQ = 'P': 输出时, 如果 INFO = 0, Q 和 IQ 紧式存储左右奇异向量, 需要 $O(N \log N)$ 的空间代替 $2*N**2$ 。

对于其他的 COMPQ 值, Q 是不被使用的。

IQ 输出参数 real 型数组, dimension (LDIQ)

其他的类似于 Q。

WORK real 型数组, dimension (LWORK)

如果, COMPQ = 'N' 则 LWORK >= (4 * N)。

如果, COMPQ = 'P' 则 LWORK >= (6 * N)。

如果, COMPQ = 'I' 则 LWORK >= (3 * N**2 + 4 * N)。

IWORK INTEGER 数组, dimension (8*N)

INFO 输出参数, INTEGER

= 0, 成功退出,

<0, 如果 INFO = -i, 则第 i 号参数值是不被接受的

>0, 算法得不到奇异值。

参数介绍完了，我们来写个程序测试一下

```
! 测试SBDSDC函数
program SBDSDC_test
  implicit none
  !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
  !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
  CHARACTER*1 UPLO ,COMPQ
  INTEGER N,LDUT,INFO ,LDU
  REAL,ALLOCATABLE ::E(:),D(:),U(:,,:),UT(:,,:),Q(:),IQ(:),WORK(:)
  REAL,ALLOCATABLE :: IWORK(:)
  UPLO = 'U'
  COMPQ = 'N'
  ! 测试就用个简单点的，3维就够了
  N = 3
  LDU = 1
  LDUT =1
  ALLOCATE(E(1:N),D(N),U(LDU,N),UT(LDUT,N),WORK(4*N),IWORK(8*N))
  ! D矩阵存储的就是要求的SVD值的矩阵主对角元素
  D(1) = 12.0
  D(2) = 15.0
  D(3) = 48.0
  ! E为其他元素，为什么就3个？先不明白就是没理解这个函数，需要在理解一下。
  E(1) = 0.0
  E(2) = 0.0
  E(3) = 0.0
  ! 调用SBDSDC例程
  CALL SBDSDC( UPLO, COMPQ, N, D, E, U, LDU, UT, LDUT, Q, IQ,WORK, IWORK, INFO )
  ! 输出一些信息
  write(*,*) INFO
  ! 这里为什么是输出D矩阵？
  write(*,*) D
end
```

当然我们不知道这样对不对。所以我们还是请出 matlab 来验证一下吧。

```
>> D = [12,0,0;0,15,0;0,0,48];
>> S = svd(D)
```

```
S =
    48
    15
    12
```



```
C:\ "E:\Fortran\SBDSDC_test\Debug\SBDSDC_test.exe"
      0
    48.00000    15.00000    12.00000
Press any key to continue
```

上个例子是一个特殊情况，那其他情况呢？

```

? 测试SBDSDC函数
program SBDSDC_test
  implicit none
  !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
  !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
  CHARACTER*1 UPLO ,COMPQ
  INTEGER N,LDUT,INFO ,LDU
  REAL,ALLOCATABLE ::E(:),D(:),U(:,,:),UT(:,,:),Q(:,),IQ(:,),WORK(:)
  REAL,ALLOCATABLE :: IWORK(:)
  UPLO = 'U'
  COMPQ = 'N'
? 测试就用个简单点的, 3维就够了
  N = 3
  LDU = 1
  LDUT =1
  ALLOCATE(E(1:N),D(N),U(LDU,N),UT(LDUT,N),WORK(4*N),IWORK(8*N))
? D矩阵存储的就是要求的SVD值的矩阵主对角元素
  D(1) = 12.0
  D(2) = 15.0
  D(3) = 48.0
? E为其他元素, 为什么就3个? 先不明白就是没理解这个函数, 需要在理解一下。
  E(1) = 2.0
  E(2) = 0.0
  E(3) = 0.0
? 调用SBDSDC例程
  CALL SBDSDC( UPLO, COMPQ, N, D, E, U, LDU, UT, LDUT, Q, IQ,WORK, IWORK, INFO )
? 输出一些信息
  write(*,*) INFO
? 这里为什么是输出D矩阵?
  write(*,*) D
end

```

```
>> D = [12, 2, 0; 0, 15, 0; 0, 0, 48];
```

```
>> S = svd(D)
```

```

S =
48.0000
15.3398
11.7342

```

```

C:\ "E:\Fortran\SBDSDC_test\Debug\SBDSDC_test.exe"
0
48.000000    15.33976    11.73421
Press any key to continue

```

2.2 SBDSQR 子程序

原型:

```

SUBROUTINE SBDSQR( UPLO, N, NCVT, NRU, NCC, D, E, VT, LDVT, U,
  $               LDU, C, LDC, WORK, INFO )

```

例程说明:

SBDSQR 是对 real 型的 $N \times N$ 阶对角矩阵 B ($B = Q * S * P^T$ (P^T 表示 P 的转置矩阵)) 进行 SVD 计算。 S 是非负数的对角矩阵 (B 的奇异值), Q 和 P 是正交矩阵。例程计算 S , 选择性的计算 $U * Q$, $P^T * VT$, 或者 $Q^T * C$ 。

参数说明:

UPLO 输入参数 (CHARACTER*1)

= 'U' : B 是上对角矩阵 (upper bidiagonal)

= 'L' : B 是下对角矩阵 (lower bidiagonal);

N 输入参数 (INTEGER)

- B 矩阵的秩。N>=0。
- NCVT 输入参数 INTEGER
VT 的列的维数，NCVT>=0。
- NRU 输入参数 INTEGER
U 的行的维数，NRU>=0。
- NCC 输入参数 INTEGER
C 的列的维数，NCC=0。
- D 输入/输出参数，real 型数组，dimension (N)
输入时，为 B 矩阵的 N 各对角元素
输出时，如果 INFO = 0，则为 B 矩阵降序的奇异值。
- E 输入/输出参数，real 型数组，dimension (N)
输入时，E 包含了要进行 SVD 的矩阵非对角元。
- VT 输入/输出参数，real 型数组，dimension (LDVT, NCVT)
输入时，为 N×NCVT 的矩阵，
输出时，VT 被重写为 P^T*VT。
如果 NCVT=0，VT 不被使用。
- LDVT 输入参数 INTEGER
VT 数组的维数。
如果 NCVT > 0，则 LDVT >= max(1,N);
如果 NCVT = 0.则 LDVT >= 1
- U 输出参数，real 型数组，dimension (LDU,N)
输入时，是 NRU×N 维矩阵，
输出时，U 被重写为 U*Q
如果 NRU = 0，则 U 不被使用。
- LDU 输入参数 INTEGER
U 的维数，LDU >= max(1,NRU)。
- C 输入/输出参数，real 型数组，dimension (LDC, NCC)
输入时，为 N×NCC 维数组
输出时，C 被重写为 Q^T*C
如果 NCC = 0，C 则不被使用。
- LDC 输入参数 INTEGER
C 的维数，
如果 NCC>0，则 LDC >= max(1,N)，
如果 NCC=0，则 LDC >=1
- WORK (workspace) real 型数组，dimension (4*N)
- INFO 输出参数，INTEGER
= 0，成功退出，
<0，如果 INFO = -i，则第 i 号参数值是不被接受的
>0，算法得不到奇异值。如果 INFO = i, E 的第 i 号元素没转换为 0。

2.3 SGEES 子程序

原型:

SUBROUTINE SGEES(JOBVS, SORT, SELECT, N, A, LDA, SDIM, WR, WI,

\$ VS, LDVS, WORK, LWORK, BWORK, INFO)

例程说明:

SGEES 例程计算 $N \times N$ real 型非对称矩阵 A 的特征值、real 型的矩阵 T (上三角矩阵), 并且选择的计算向量 Z, 使 $A = Z * T * (Z^{**}T)$ 成立。

参数说明:

JOBVS **输入参数** CHARACTER*1

* = 'N': 不计算因式分解的向量;

* = 'V': 计算因式分解的向量。

* SORT **输入参数** CHARACTER*1

* 指定特征值是否构成在对角矩阵的分解量中。

* = 'N': 特征值不被指定;

* = 'S': 特征值被指定

* SELECT **输入参数** LOGICAL FUNCTION of two REAL arguments

* SELECT must be declared EXTERNAL in the calling subroutine.

* 如果 SORT = 'S', SELECT 被用来选择特征值储存在分解形式的顶部。

* 如果 SORT = 'N', SELECT 不被使用

* 如果 SELECT(WR(j),WI(j)) 是 true, 则特征值 $WR(j) + \sqrt{-1} * WI(j)$ 被选择

* N **输入参数** INTEGER

* 矩阵 A 的维数. $N \geq 0$.

* A **输入/输出参数** REAL 数组, dimension (LDA,N)

* 输入时, A[N][N].

* 输出时, T.

* LDA **输入参数** INTEGER

* 决定数组 A 的尺寸. $LDA \geq \max(1, N)$.

* SDIM **输出参数** INTEGER

* 如果 SORT = 'N', 则 SDIM = 0.

* 如果 SORT = 'S', 则 SDIM = 特征值的个数 (after sorting)

* SELECT 必须是 true.

* WR **输出参数** REAL 数组 dimension (N)

* WI **输出参数** REAL 数组 dimension (N)

* VS **输出参数** REAL 数组, dimension (LDVS,N)

* 如果 JOBVS = 'V', VS 是正交矩阵 Z

* 如果 JOBVS = 'N', VS 不被使用

* LDVS **输入参数** INTEGER

* 决定数组 VS 的维数. $LDVS \geq 1$;

* 如果 JOBVS = 'V', 则 $LDVS \geq N$.

* WORK (workspace/output) REAL 数组, dimension (LWORK)

* 输出时, if INFO = 0, WORK(1) 包含最优的 LWORK.

* LWORK **输入参数** INTEGER

* 数组 WORK 的尺寸. $LWORK \geq \max(1, 3 * N)$.

* 为了得到较好的结果, LWORK 必须要大。

* BWORK (workspace) LOGICAL 数组, dimension (N)

* 如果 SORT = 'N', 不被使用。

*

- * INFO **输出参数** INTEGER
- * = 0:成功退出
- * < 0: 如果 INFO = -i, 第 i 号元素不被接受。
- * > 0:如果 INFO = i, 并且 i
- * <= N: QR 算法计算失败

2.4 SGEEV 程序

原型:

**SUBROUTINE SGEEV(JOBVL, JOBVR, N, A, LDA, WR, WI, VL, LDVL, VR,
\$ LDVR, WORK, LWORK, INFO)**

例程说明:

SGEEV 计算非对称矩阵 A (N*N) 的特征值, 选择性的左和/或右计算特征向量。A 矩阵的右特征向量满足:

$$A * V(j) = \lambda(j) * V(j), \text{ 其中 } \lambda(j) \text{ 是特征值。}$$

A 矩阵的左特征向量满足:

$$u(j) **H * A = \lambda(j) * u(j) **H, \text{ 其中 } u(j) **H \text{ 为 } u(j) \text{ 的共轭变换。}$$

参数说明:

JOBVL **输入参数** CHARACTER*1

= 'N': left eigenvectors of A are not computed;

= 'V': left eigenvectors of A are computed.

JOBVR **输入参数** CHARACTER*1

= 'N': right eigenvectors of A are not computed;

= 'V': right eigenvectors of A are computed.

N **输入参数** INTEGER

The order of the matrix A. N >= 0.

A **输入/输出参数** REAL array, dimension (LDA,N)

On entry, the N-by-N matrix A.

On exit, A has been overwritten.

LDA **输入参数** INTEGER

The leading dimension of the array A. LDA >= max(1,N).

WR **输出参数** REAL array, dimension (N)

WI **输出参数** REAL array, dimension (N)

WR and WI contain the real and imaginary parts, respectively, of the computed eigenvalues. Complex conjugate pairs of eigenvalues appear consecutively

with the eigenvalue having the positive imaginary part first.

VL **输出参数** REAL array, dimension (LDVL,N)

If JOBVL = 'V', the left eigenvectors $u(j)$ are stored one after another in the columns of VL, in the same order as their eigenvalues.

If JOBVL = 'N', VL is not referenced.

If the j -th eigenvalue is real, then $u(j) = VL(:,j)$, the j -th column of VL.

If the j -th and $(j+1)$ -st eigenvalues form a complex conjugate pair, then $u(j) = VL(:,j) + i*VL(:,j+1)$ and $u(j+1) = VL(:,j) - i*VL(:,j+1)$.

LDVL **输入参数** INTEGER

The leading dimension of the array VL. LDVL ≥ 1 ; if JOBVL = 'V', LDVL $\geq N$.

VR **输出参数** REAL array, dimension (LDVR,N)

If JOBVR = 'V', the right eigenvectors $v(j)$ are stored one after another in the columns of VR, in the same order as their eigenvalues.

If JOBVR = 'N', VR is not referenced.

If the j -th eigenvalue is real, then $v(j) = VR(:,j)$, the j -th column of VR.

If the j -th and $(j+1)$ -st eigenvalues form a complex conjugate pair, then $v(j) = VR(:,j) + i*VR(:,j+1)$ and $v(j+1) = VR(:,j) - i*VR(:,j+1)$.

LDVR **输入参数** INTEGER

The leading dimension of the array VR. LDVR ≥ 1 ; if JOBVR = 'V', LDVR $\geq N$.

WORK **工作/输出参数** REAL array, dimension (LWORK)

On exit, if INFO = 0, WORK(1) returns the optimal LWORK.

LWORK **输入参数** INTEGER

The dimension of the array WORK. LWORK $\geq \max(1, 3*N)$, and if JOBVL = 'V' or JOBVR = 'V', LWORK $\geq 4*N$. For good performance, LWORK must generally be larger.

If LWORK = -1, then a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error

message related to LWORK is issued by XERBLA.

INFO **输出参数** INTEGER

= 0: successful exit

< 0: if INFO = -i, the i-th argument had an illegal value.

> 0: if INFO = i, the QR algorithm failed to compute all the
 eigenvalues, and no eigenvectors have been computed;
 elements i+1:N of WR and WI contain eigenvalues which
 have converged.

2.5 SGELQF 例程

原型:

SUBROUTINE SGELQF(M, N, A, LDA, TAU, WORK, LWORK, INFO)

例程说明:

SGELQF 计算矩阵 $A(M \times N)$ 的 LQ 分解: $A = L * Q$ 。

参数说明:

M 输入参数, INTEGER

 矩阵 A 的行数。 $M \geq 0$

N 输入参数, INTEGER

 矩阵 A 的列数。 $N \geq 0$

A 输入/输出参数, real 数组。 dimension (LDA,N)

 输入时, 为 $M \times N$ 的矩阵 A

 输出时, 包含如下元素: 矩阵 L; 对角线以上的元素, 含数组 TAU 代表正交矩阵 Q

LDA 输入参数, INTEGER

 决定着数组 A 的尺寸, $LDA \geq \max(1, M)$

TAU 输出参数, real 数组 dimension (min(M,N))

 包含 Q。

WORK 工作间/输出参数, real 数组, dimension (LWORK)

 输出时, 如果 INFO = 0, WORK(1) 返回最优的 LWORK。

LWORK 输入参数, INTEGER

 决定数组 WORK 尺寸。 $LWORK \geq \max(1, M)$

 为最优的性能, $LWORK \geq M * NB$, NB 是最优的分块大小。

INFO 输出参数, INTEGER

= 0: 成功退出

< 0: 如果 INFO = -i, 则第 i 号参数是不可接受的数值。

! 对SGELQF函数进行代码演示

```
program SGELQF_TEST
  implicit none
  !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
  !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
  INTEGER M,N,LDA,LWORK,INFO
  REAL,ALLOCATABLE::A(:,,:),TAU(:),WORK(:)
  M = 4
  N = 3
  LDA = 4
  LWORK = 4
  ALLOCATE(A(LDA,N),TAU(MIN(M,N)),WORK(LWORK))
  A(1,1) = 1.1
  A(1,2) = 2.2
  A(1,3) = 3.3
  A(2,1) = 4.4
  A(2,2) = 5.5
  A(2,3) = 6.6
  A(3,1) = 7.7
  A(3,2) = 8.8
  A(3,3) = 9.9
  A(4,1) = 10.7
  A(4,2) = 11.7
  A(4,3) = 12.4
  CALL SGELQF( M, N, A, LDA, TAU, WORK, LWORK, INFO )
  WRITE(*,*) A
  WRITE(*,*) INFO
  WRITE(*,*) TAU
end program SGELQF_TEST
```

```
"E:\FORTRAN\SGELQF_TEST\Debug\SGELQF_TEST.exe"
-4.115823      -9.407595      -14.69937      -19.05573      0.4217934
 2.160356       4.320714       6.481071       0.6326901      0.8597679
 5.0079859E-07 -0.1224733
 0
 1.267261      1.149953      0.0000000E+00
Press any key to continue
```

2.6 SGELQ2 例程

原型:

SUBROUTINE SGELQ2(M, N, A, LDA, TAU, WORK, INFO)

例程说明:

SGELQ2 计算矩阵 $A(M \times N)$ 的 LQ 分解; $A = L * Q$ 。

参数说明:

- M 输入参数, INTEGER
矩阵 A 的行数。 $M \geq 0$
- N 输入参数, INTEGER
矩阵 A 的列数。 $N \geq 0$
- A 输入/输出参数, real 数组。 dimension (LDA,N)
输入时, 为 $M \times N$ 的矩阵 A

输出时, 包含如下元素: 矩阵 L; 对角线以上的元素, 含数组 TAU 代表正交矩阵 Q

LDA 输入参数, INTEGER

决定着数组 A 的尺寸, $LDA \geq \max(1, M)$

TAU 输出参数, real 数组 dimension (min(M,N))

包含 Q。

WORK 工作间/输出参数, real 数组, dimension (M)

INFO 输出参数, INTEGER

= 0: 成功退出

< 0: 如果 INFO = -i, 则第 i 号参数是不可接受的数值。

! 对 SGELQF 函数进行代码演示

```

program SGELQF_TEST
  implicit none
  !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
  !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
  INTEGER M,N,LDA,LWORK,INFO
  REAL,ALLOCATABLE::A(:, :),TAU(:, :),WORK(:)
  M = 4
  N = 3
  LDA = 4
  LWORK = 4
  ALLOCATE(A(LDA,N),TAU(MIN(M,N)),WORK(LWORK))
  A(1,1) = 1.1
  A(1,2) = 2.2
  A(1,3) = 3.3
  A(2,1) = 4.4
  A(2,2) = 5.5
  A(2,3) = 6.6
  A(3,1) = 7.7
  A(3,2) = 8.8
  A(3,3) = 9.9
  A(4,1) = 10.7
  A(4,2) = 11.7
  A(4,3) = 12.4
  ! CALL SGELQF( M, N, A, LDA, TAU, WORK, LWORK, INFO )
  CALL SGELQ2( M, N, A, LDA, TAU, WORK, INFO )
  WRITE(*,*) A
  WRITE(*,*) INFO
  WRITE(*,*) TAU
end program SGELQF_TEST

```

```

C:\> "E:\FORTRAN\SGELQF_TEST\Debug\SGELQF_TEST.exe"
-4.115823      -9.407595      -14.69937      -19.05573      0.4217934
 2.160356       4.320714       6.481071       0.6326901      0.8597679
 5.0079859E-07 -0.1224733
              0
 1.267261      1.149953      0.0000000E+00
Press any key to continue

```

2.7 SGELS 例程

原型:

**SUBROUTINE SGELS(TRANS, M, N, NRHS, A, LDA, B, LDB, WORK, LWORK,
\$ INFO)**

例程说明:

SGELS 解决关于 $M \times N$ 的矩阵 A 或 A 的转置或者是 A 的 QR/LQ 分解形式的超定线性方程组或欠定线性方程组。A 是满秩的。

例如:

- 1) 如果 TRANS = 'N' 并且 $m \geq n$: 计算最小二乘法的线性超定方程组, $\text{minimize} \|B - A * X\|$
- 2) 如果 TRANS = 'N' 并且 $m < n$: 以最小准则解决一个欠定线性方程组, $A * X = B$ 。
- 3) 如果 TRANS = 'T' 并且 $m \geq n$: 以最小准则解决一个欠定线性方程组, $A ** T * X = B$ 。
- 4) 如果 TRANS = 'T' 并且 $m < n$: 计算最小二乘法的线性超定方程组, $\text{minimize} \|B - A ** T * X\|$ 。

参数说明:

TRANS 输入参数, CHARACTER

= 'N': 引用矩阵 A 的线性系统

= 'T': 引用矩阵 $A ** T$ 的线性系统

M 输入参数, INTEGER

矩阵 A 的行数, $M \geq 0$ 。

N 输入参数, INTEGER

矩阵 A 的列数, $N \geq 0$ 。

NRHS 输入参数, INTEGER

右侧尺寸, 如: B 和 X 的列数, $NRHS \geq 0$ 。

A 输入/输出参数, real 数组, dimension (LDA,N)

输入时, $M \times N$ 的矩阵 A;

输出时, 如果 $M \geq N$, A 被 SGEQRF 返回的 QR 因式详细重写。

如果 $M < N$, A 被 SGELQF 返回的 LQ 因式详细重写。

LDA 输入参数, INTEGER

决定数组 A 的尺寸, $LDA \geq \max(1, M)$ 。

B 输入/输出参数, dimension (LDB, NRHS)

输入时, 矩阵 B 的右侧向量, 并且是列式存储。

如果 TRANS = 'N' 并且 $m \geq n$, B 的第 1 行到第 n 行包含最小二乘解向量;

LDB 输入参数, INTEGER

决定数组 B 的尺寸, $LDB \geq \max(1, M, N)$

WORK 工作间/输出参数, real 数组 dimension (LWORK)

输出时, 如果 INFO = 0, WORK(1) 返回最优的 LWORK。

LWORK 输入参数, INTEGER

决定 WORK 的尺寸。 $LWORK \geq \max(1, MN + \max(MN, NRHS))$ 。

INFO 输出参数, INTEGER

= 0, 成功退出

= -i, 第 i 号参数不可接受。

更多内容请访问 <http://www.feachina.com> 输电与有限元技术
论坛

