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，上海森博格矩阵
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×N 阶对角矩阵 B（B＝U＊S＊VT）进行 SVD 计算。

【注：奇异值分解（sigular 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（INTEGERT）

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

```fortran
!   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,LDVT,INFO ,LDU
        REAL,ALLOCATABLE ::E(:),D(:),U(:,:),VT(:,:),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
        LDVT =1
        ALLOCATE(E(1:N),D(N),U(LDU,N),VT(LDVT,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, VT, LDVT, 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
```

```
"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？

```fortran
!   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,LDVT,INFO ,LDU
        REAL,ALLOCATABLE ::E(:),D(:),U(:,:),VT(:,:),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
        LDVT =1
        ALLOCATE(E(1:N),D(N),U(LDU,N),VT(LDVT,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, VT, LDVT, 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
```

```
"E:\Fortran\SBDSDC_test\Debug\SBDSDC_test.exe"
           0
   48.00000        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×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×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 >= 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 >= 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 c 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 >= 1;
* 如果 JOBVS = 'V',则 LDVS >= 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 >= 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） = λ(j) *V（j），其中λ(j)是特征值。

A 矩阵的左特征向量满足：

u（j）**H*A = λ(j) *u（j）**H，其中 u（j）**H 为 u（j）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 >= 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 >= 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 >= max(1,3*N), and if JOBVL = 'V' or JOBVR = 'V', LWORK >= 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×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>=0

N 输入参数，INTEGER

矩阵 A 的列数。N>=0

A 输入/输出参数，real 数组。dimension (LDA,N)

输入时，为 M×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>=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 >= max(1,M)

为最优的性能，LWORK>=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×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>=0

N　输入参数，INTEGER

　　矩阵 A 的列数。N>=0

A　输入/输出参数，real 数组。dimension (LDA,N)

　　输入时，为 M×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>=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。

```fortran
!    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
```

```
"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×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>=n：计算最小二乘法的线性超定方程组， 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>=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，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>=0。
N 输入参数,INTEGER
矩阵 A 的列数，N>=0。
NRHS 输入参数，INTEGER
右侧尺寸，如：B 和 X 的列数，NRHS>=0。
A 输入/输出参数，real 数组，dimension (LDA,N)
输入时，M×N 的矩阵 A;
输出时，如果 M>=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 >= 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 >= 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 >= 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 >= 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。

2.8 SGELSS 例程

原型：

SUBROUTINE SGELSS(M, N, NRHS, A, LDA, B, LDB, S, RCOND, RANK, WORK, LWORK, INFO)

例程说明：

SGELSS 最小规模解决线性方程组的最小二乘方解。

Minimize 2-norm(| b - A*x |)。应用矩阵 A 的 SVD。

A 是 M×N 的矩阵。

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>=0。

N 输入参数,INTEGER

矩阵 A 的列数，N>=0。

NRHS 输入参数，INTEGER

右侧尺寸，如：B 和 X 的列数，NRHS>=0。

A 输入/输出参数，real 数组，dimension (LDA,N)

输入时，M×N 的矩阵 A；

输出时，如果 M>=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 >= 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 >= 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 >= MAX(1,M,N)

S 输出参数，real 数组，dimension (min(M,N))

一个降序的奇异值。在 2-norm = S(1)/S(min(m,n))中的条件数值。

RCOND 输入参数，REAL

矩阵 A 有效地秩，例如：较 RCOND*S(1)大的奇异值的个数。

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。为了优越的性能，LWORK 应该较大。

INFO 输出参数，INTEGER

= 0,成功退出

= -i，第 i 号参数不可接受。

2.9 SGEQL2 例程

原型：

SUBROUTINE SGEQL2(M, N, A, LDA, TAU, WORK, INFO)

例程说明：

SGEQL2 对 M×N 阶矩阵 A 进行 QL 分解。A = Q*L

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入时，为 M×N 的矩阵 A。

输出时，如果 m>=n 时，下三角子矩阵 A(m-n+1:m,1:n)包含 N×N 阶下三角矩阵 L；

如果 m<=n，元素在第(n-m)对角线上或下包含 m×n 阶以下下矩阵 L。余下的元素存放在 TAU 数组里。

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

TAU 输出参数，REAL 数组 dimension (min(M,N))

存放剩余的元素。

WORK 工作间，REAL 数组，dimension (N)

INFO 输出参数，INTEGER

= 0：成功退出

< 0：如果，INFO = -i，第 i 号参数无

2.10 SGEQLF 例程

原型：

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

例程说明：

SGEQLF 对 M×N 阶矩阵 A 进行 QL 分解。A = Q*L

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入时，为 M×N 的矩阵 A。

输出时，如果 m>=n 时，下三角子矩阵 A(m-n+1:m,1:n)包含 N×N 阶下三角矩阵 L；

如果 m<=n，元素在第(n-m)对角线上或下包含 m×n 阶以下矩阵 L。余下的元素存放在 TAU 数组里。

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

TAU 输出参数，REAL 数组 dimension (min(M,N))

存放剩余的元素。

WORK 工作间，REAL 数组，dimension (LWORK)

LWORK 输入参数，决定 WORK 的尺寸。

INFO 输出参数，INTEGER

= 0：成功退出

< 0：如果，INFO = -i，第 i 号参数无效。

```
!    对SGEQLF函数进行代码演示
     program SGEQLF_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 SGEQLF_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
```

2.11 SGEQR2 例程

原型：

SUBROUTINE SGEQR2(M, N, A, LDA, TAU, WORK, INFO)

例程说明：

SGEQR2 对 M×N 阶矩阵 A 进行 QR 分解。A = Q*R

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入时，为 M×N 的矩阵 A。

输出时，如果 m>=n 时，下三角子矩阵 A(m-n+1:m,1:n)包含 N×N 阶下三角矩阵 L；

如果 m<=n，元素在第(n-m)对角线上或下包含 m×n 阶以下下矩阵 L。余下的元素存放在 TAU 数组里。

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

TAU 输出参数，REAL 数组 dimension (min(M,N))

存放剩余的元素。

WORK 工作间，REAL 数组，dimension (N)

INFO 输出参数，INTEGER

= 0：成功退出

< 0：如果，INFO = -i，第 i 号参数无效。

2.12 SGEQRF 例程

原型：

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

例程说明：

SGEQRF 对 M×N 阶矩阵 A 进行 QR 分解。A = Q*R

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入时，为 M×N 的矩阵 A。

输出时，如果 m>=n 时，下三角子矩阵 A(m-n+1:m,1:n)包含 N×N 阶下三角矩阵 L；

如果 m<=n，元素在第(n-m)对角线上或下包含 m×n 阶以下下矩阵 L。余下的元素存放在 TAU 数组里。

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

TAU 输出参数，REAL 数组 dimension (min(M,N))

存放剩余的元素。

WORK 工作间，REAL 数组，dimension (LWORK)

LWORK 输入参数，决定 WORK 的尺寸。

INFO 输出参数，INTEGER

= 0：成功退出

< 0：如果，INFO = -i，第 i 号参数无效。

2.13 SGERQ2 例程

原型：

SUBROUTINE SGERQ2(M, N, A, LDA, TAU, WORK, INFO)

例程说明：

SGERQ2 对 M×N 阶矩阵 A 进行 RQ 分解。A = R*Q

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入时，为 M×N 的矩阵 A。

输出时，如果 m>=n 时，下三角子矩阵 A(m-n+1:m,1:n)包含 N×N 阶下三角矩阵 L；

如果 m<=n，元素在第(n-m)对角线上或下包含 m×n 阶以下下矩阵 L。余下的元素存放在 TAU 数组里。

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

TAU 输出参数，REAL 数组 dimension (min(M,N))

存放剩余的元素。

WORK 工作间，REAL 数组，dimension (N)

INFO 输出参数，INTEGER

= 0：成功退出

< 0：如果，INFO = -i，第 i 号参数无效。

2.14 SGERQF 例程

原型：

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

例程说明：

SGERQF 对 M×N 阶矩阵 A 进行 RQ 分解。A = R*Q

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入时，为 M×N 的矩阵 A。

输出时，如果 m>=n 时，下三角子矩阵 A(m-n+1:m,1:n)包含 N×N 阶下三角矩阵 L；

如果 m<=n，元素在第(n-m)对角线上或下包含 m×n 阶以下矩阵 L。余下的元素存放在 TAU 数组里。

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

TAU 输出参数，REAL 数组 dimension (min(M,N))

存放剩余的元素。

WORK 工作间，REAL 数组，dimension (LWORK)

LWORK 输入参数，决定 WORK 的尺寸。

INFO 输出参数，INTEGER

= 0：成功退出

< 0：如果，INFO = -i，第 i 号参数无效。

2.15 SLACPY 例程

原型：

SUBROUTINE SLACPY(UPLO, M, N, A, LDA, B, LDB)

例程说明：

SLACPY 拷贝二维数组 A 的所有或部分到另一个矩阵 B。

参数说明：

UPLO 输入参数，CHARACTER*1

指定 A 哪一部分拷贝到 B 中。

= 'U':上三角部分

= 'L':下三角部分

M 输入参数，INTEGER

矩阵 A 的行数，M>=0.

N 输入参数，INTEGER

矩阵 A 的列数，N>=0

A 输入参数，REAL 数组，dimension (LDA,N)

LDA 输入参数，INTEGER

决定矩阵 A 的尺寸，LDA >= max(1,M)。

B 输出参数，REAL 数组 dimension (LDB,N)

输出时，B = 在 UPLO 中描述的 A 的部分或整体。

LDB 输入参数，INTEGER

决定数组 B 的尺寸，LDB >= max(1,M)。

我们举一个例子：

A 矩阵如下：

1.0	2.0	3.0
4.0	5.0	6.0
7.0	8.0	9.0

```fortran
    program Test00
        implicit none
        !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
        !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
!       SLACPY函数演示
        CHARACTER*1 UPLO
        INTEGER M,N,LDA,LDB
        REAL,ALLOCATABLE:: A(:,:)
        REAL B(3,3)
        UPLO = 'L'
        M = 3
        N = 3
        LDA = 3
        LDB = 3
        ALLOCATE(A(LDA,N))
        A(1,1) = 1.0
        A(1,2) = 2.0
        A(1,3) = 3.0
        A(2,1) = 4.0
        A(2,2) = 5.0
        A(2,3) = 6.0
        A(3,1) = 7.0
        A(3,2) = 8.0
        A(3,3) = 9.0
        DATA B /0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0/
        CALL SLACPY( UPLO, M, N, A, LDA, B, LDB )
        WRITE(*,*) '输出矩阵A，看看是怎么存储的'
        WRITE(*,*) A
        WRITE(*,*) 'SLACPY函数演示'
        WRITE(*,*) B
    end program Test00
```

```
"E:\FORTRAN\Test00\Debug\Test00.exe"
 输出矩阵A，看看是怎么存储的
   1.000000       4.000000       7.000000       2.000000       5.000000
   8.000000       3.000000       6.000000       9.000000
 SLACPY函数演示
   1.000000       4.000000       7.000000     0.0000000E+00    5.000000
   8.000000     0.0000000E+00  0.0000000E+00    9.000000
Press any key to continue
```

2.16 SLAED0 例程

原型：

SUBROUTINE SLAED0(ICOMPQ, QSIZ, N, D, E, Q, LDQ, QSTORE, LDQS, WORK, IWORK, INFO)

例程说明：

SLAED0 使用分治法计算对称三角线矩阵所有的特征值及其对应的特征向量。

参数说明：

ICOMPQ 输入参数，INTEGER

= 0：只计算特征向量；

= 1：同时计算原密集对称矩阵的特征向量。输出时，Q 包含将原矩阵化为对 角矩阵的正交矩阵。

= 2：计算对角矩阵的特征值与特征向量。

QSIZ 输入参数，INTEGER

决定将原矩阵化为三角矩阵的正交矩阵的尺寸。如果 ICOMPQ = 1，则 QSIZ >= N。

N 输入参数，INTEGER

决定对称三角矩阵的尺寸。N>=0

D 输入/输出参数，REAL 数组，dimension (N)

输入时，为三角矩阵的主对角线。

输出时，为特征向量。

E 输入参数，REAL 数组，dimension (N-1)

对角矩阵的非对角元素。

输出时，E 被破坏。

Q 输入/输出参数，REAL 数组，dimension (LDQ, N)

输入时，Q 包含了 N×N 的正交矩阵。

如果 ICOMPQ = 0，Q 不被使用。

如果 ICOMPQ = 1，输出时...

如果 ICOMPQ = 2，输出时 Q 包含了特征向量。

LDQ 输入参数，INTEGER

决定 Q 的尺寸，如果希望得到特征向量，则 LDQ >= max (1,N)。在任何情况下 LDQ >= 1 恒成立。

QSTORE 工作间，REAL 数组，dimension (LDQS, N)

只有在 ICOMPQ = 1 时使用。

LDQS 输入参数，INTEGER

决定 QSTORE 数组的维数。如果 ICOMPQ = 1，则 LDQS >= max(1,N)。在任何情 况下， LDQS >= 1 恒成立。

WORK 工作间，REAL 数组。

如果 ICOMPQ = 0 或 1，WORK 的尺寸至少是 1 + 3*N + 2 *N*lg N + 2*N**2。

如果 ICOMPQ =2，则 WORK 的尺寸至少是 4*N + N**2。

IWORK 工作间，INTEGER 数组

如果 ICOMPQ = 0 或 1，IWORK 的尺寸至少是 6 + 6*N + 5 *N*lg N。

如果 ICOMPQ = 2，IWORK 的尺寸至少是 3 + 5*N。

INFO 输出参数，INTEGER

= 0：成功退出，

<0：如果 INFO =-i，则第 i 号参数不可接受。

>0：算法不能计算特征值。

我们来测试一下：

```
!    对函数SLAED0进行代码测试
     program SLAED0_test
         implicit none
         !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
         !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
         INTEGER ICOMPQ,QSIZ,N,LDQ,LDQS,INFO
         INTEGER,ALLOCATABLE::IWORK(:)
         REAL,ALLOCATABLE::D(:),E(:),Q(:,:),QSTORE(:,:),WORK(:)
         ICOMPQ = 0
         QSIZ = 1
         N = 3
         LDQ = MAX(1,N)
         LDQS= MAX(1,N)
         ALLOCATE(D(N),E(N-1),Q(LDQ,N),QSTORE(LDQS,N),WORK(100),IWORK(100))
         D(1) = 10.1
         D(2) = 21.2
         D(3) = 2.1
         E(1) = 1.0
         E(2) = 2.0
         Q = 0
         CALL  SLAED0( ICOMPQ, QSIZ, N, D, E, Q, LDQ, QSTORE, LDQS, WORK, IWORK, INFO )
         WRITE(*,*) D
     end program SLAED0_test
```

```
"E:\FORTRAN\SLAED0_test\Debug\SLAED0_test.exe"
   1.891522        10.01446        21.49402
Press any key to continue
```

```fortran
!    对函数SLAED0进行代码测试
     program SLAED0_test
         implicit none
         !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
         !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
         INTEGER ICOMPQ,QSIZ,N,LDQ,LDQS,INFO
         INTEGER,ALLOCATABLE::IWORK(:)
         REAL,ALLOCATABLE::D(:),E(:),Q(:,:),QSTORE(:,:),WORK(:)
         ICOMPQ = 0
         QSIZ = 1
         N = 3
         LDQ = MAX(1,N)
         LDQS= MAX(1,N)
         ALLOCATE(D(N),E(N-1),Q(LDQ,N),QSTORE(LDQS,N),WORK(100),IWORK(100))
         D(1) = 10.1
         D(2) = 21.2
         D(3) = 2.1
         E(1) = 0.0
         E(2) = 0.0
         Q = 0
         CALL  SLAED0( ICOMPQ, QSIZ, N, D, E, Q, LDQ, QSTORE, LDQS, WORK, IWORK, INFO )
         WRITE(*,*) D
     end program SLAED0_test
```

```
"E:\FORTRAN\SLAED0_test\Debug\SLAED0_test.exe"
   2.100000        10.10000        21.20000
Press any key to continue
```

2.17 SGEHRD 例程

原型：

SUBROUTINE SGEHRD(N, ILO, IHI, A, LDA, TAU, WORK, LWORK, INFO)

例程说明：

SGEHRD 通过正交相似变换将矩阵 A 转换为 H，Q' * A * Q = H

参数说明：

N 输入参数，INTEGER

矩阵 A 的维数。N>=0

ILO 输入参数，INTEGER

IHI 输入参数，INTEGER

假设矩阵 A 在 1 到 ILO-1 行和 IHI+1 到 N 已经是一个上三角矩阵。

ILO 和 IHI 通常是通过以前调用 SGEBAL 函数设置；否则要从 1 到 N 一一设置。

如果 N > 0，则 1 <= ILO <= IHI <= N；如果 N=0,则 ILO=1 并且 and IHI=0.

A 输入/输出参数，REAL 数组，dimension (LDA,N)

输入时，需要转换的 N×N 矩阵。

输出时，A 的上三角和副对角线被 Hessenberg 矩阵 H 重写，并且在主对角线以下的元 素放在 TAU 数组中，代表正交矩阵 Q。

LDA 输入参数，INTEGER

决定矩阵 A 尺寸，LDA >= max(1,N)。

TAU 输出参数，REAL 数组，dimension (N-1)。

TAU 的元素 (1 到 ILO-1 和 IHI 到 N-1)被设置为 0。

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

数组 WORK 的长度， LWORK >= max(1,N)。为了最优的

性能，LWORK >= N*NB， NB 是最优的分块矩阵。

INFO 输出参数，INTEGER

= 0:成功退出

<=0:如果 INFO = -i，第 i 个参数是不被接受的。

2.18 SGEHD2 例程

原型：

SUBROUTINE SGEHD2(N, ILO, IHI, A, LDA, TAU, WORK, INFO)

例程说明：

SGEHD2 通过正交相似变换将矩阵 A 转换为 H，Q' * A * Q = H

参数说明：

N 输入参数，INTEGER

矩阵 A 的维数。N>=0

ILO 输入参数，INTEGER

IHI 输入参数，INTEGER

假设矩阵 A 在 1 到 ILO-1 行和 IHI+1 到 N 已经是一个上三角矩阵。

ILO 和 IHI 通常是通过以前调用 SGEBAL 函数设置；否则要从 1 到 N 一一设置。

如果 N > 0，则 1 <= ILO <= IHI <= N; 如果 N=0,则 ILO=1 并且 and IHI=0.

A 输入/输出参数，REAL 数组，dimension (LDA,N)

输入时，需要转换的 N×N 矩阵。

输出时，A 的上三角和副对角线被 Hessenberg 矩阵 H 重写，并且在主对角线以下的元 素放在 TAU 数组中，代表正交矩阵 Q。

LDA 输入参数，INTEGER

决定矩阵 A 尺寸，LDA >= max(1,N)。

TAU 输出参数，REAL 数组，dimension (N-1)。

TAU 的元素 (1 到 ILO-1 和 IHI 到 N-1)被设置为 0。

WORK 工作间/输出参数，REAL 数组，dimension (N)

INFO 输出参数，INTEGER

= 0:成功退出

<=0:如果 INFO = -i，第 i 个参数是不被接受的。

```
The contents of A are illustrated by the following example, with
n = 7, ilo = 2 and ihi = 6:

on entry,                        on exit,

( a   a   a   a   a   a   a )    (  a   a   h   h   h   h   a )
(     a   a   a   a   a   a )    (      a   h   h   h   h   a )
(     a   a   a   a   a   a )    (      h   h   h   h   h   h )
(     a   a   a   a   a   a )    (      v2  h   h   h   h   h )
(     a   a   a   a   a   a )    (      v2  v3  h   h   h   h )
(     a   a   a   a   a   a )    (      v2  v3  v4  h   h   h )
(                         a )    (                          a )
```

```fortran
program SGEHD2_test
    implicit none
    !DEC$OBJCOMMENT LIB :'lapack_win32.lib'
    !DEC$OBJCOMMENT LIB :'blas_win32.lib'
    INTEGER N,ILO,IHI,LDA,INFO
    REAL,ALLOCATABLE::A(:,:),TAU(:),WORK(:)
    N = 4
    ILO = 1
    IHI = 4
    LDA =MAX(1,N)
    ALLOCATE(A(LDA,N),TAU(N-1),WORK(N))
    A = RESHAPE((/10.0,11.2,13.2,14.3,&
                  10.9,78.1,36.2,16.7,&
                  21.2,26.3,45.3,98.3,&
                  45.5,63.9,32.5,74.6,15.6/),(/4,4/))
    WRITE(*,*) '输出矩阵A-------'
    WRITE(*,10) A
    WRITE(*,*) '----------------A'
    CALL SGEHD2( N, ILO, IHI, A, LDA, TAU, WORK, INFO )
    WRITE(*,20) A
    WRITE(*,*) '输出矩阵TAU--'
    WRITE(*,20) TAU
10  FORMAT(/4X,F4.1,4X,F4.1,4X,F4.1,4X,F4.1)
20  FORMAT(/4X,F,4X,F,4X,F,4X,F)
end program SGEHD2_test
```

```
"E:\V-FORTRAN\SGEHD2_test\Debug\SGEHD2_test.exe"
 输出矩阵A-------

    10.0    11.2    13.2    14.3

    10.9    78.1    36.2    16.7

    21.2    26.3    45.3    98.3

    45.5    63.9    32.5    74.6
 ----------------A

         10.0000000        -22.4537296          0.3922299          0.4249158

        -46.8772888        158.2473907        -34.5106277         -0.0871722

          5.0648141         12.1008596          8.1621161        -56.9869385

         20.3805637         -3.2758701          9.8870344         31.5904713
 输出矩阵TAU--

          1.4988035          1.9849167          0.0000000
Press any key to continue
```

2.20 STZRZF 例程

原型：

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

例程说明：

STZRZF 例程通过正交变换将上梯形矩阵 A（M×N，M <=N）变换为上三角矩阵。上梯形矩阵 A 被分解为:

A = （R 0）*Z,

其中：Z 是 N×N 接正交矩阵，R 是 M×M 上三角矩阵。

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0。

N 输入参数，INTEGER

矩阵 A 的列数，N>=0。

A 输入输出参数，REAL 数组，dimension (LDA,N)

输入时，包含要分解的上梯形矩阵。

输出时，包含上三角矩阵，还有其余的 A 的元素。

LDA 输入参数，INTEGER

数组 A 的尺寸， LDA >= max(1,M)

TAU 输入参数，REAL 数组，dimension (M)

分解因子

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 的尺寸， LWORK >= max(1,M)。

为了最优的性能，LWORK >= M*NB，NB 是最优的分块尺寸。

INFO 输出参数，INTEGER

= 0：成功退出

<0：如果 INFO=-i,则第 i 号参数不被接受。

```fortran
!    对STZRZF例程进行代码演示
    program STZRZF_test
        implicit none
        !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
        !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
        INTEGER M,N,LDA,LWORK,INFO
        REAL  ,ALLOCATABLE::A(:,:),WORK(:),TAU(:)
        M = 3
        N =4
        LDA = M
        LWORK = M
        ALLOCATE(A(LDA,N),TAU(M),WORK(LWORK))
        A = RESHAPE((/1.0,1.2,1.3,1.4,&
                      2.1,2.2,2.3,2.4,&
                      3.1,3.2,3.3,3.5/),(/3,4/))
        WRITE(*,*) '输出变化之前的A矩阵'
        WRITE(*,10) A
        CALL STZRZF( M, N, A, LDA, TAU, WORK, LWORK, INFO )
        WRITE(*,*) '输出变化之后的A矩阵'
        WRITE(*,10) A
10      FORMAT(/4X,F,4X,F,4X,F,4X,F)
    end program STZRZF_test
```

M×M上三角矩

```
"E:\FORTRAN\STZRZF_test\Debug\STZRZF_test.exe"
 输出变化之前的A矩阵

         1.0000000          1.2000000          1.3000000          1.4000000

         2.0999999          2.2000000          2.3000000          2.4000001

         3.0999999          3.2000000          3.3000000          3.5000000
 输出变化之后的A矩阵

        -1.0092962          1.2000000          1.3000000         -1.4495823

        -2.1361642          2.2000000         -3.9204633         -4.0616260

        -4.6754680          0.0680189          0.0923960          0.4501337
Press any key to continue
```

2.21 STZRQF 例程

原型：

SUBROUTINE STZRQF(M, N, A, LDA, TAU, INFO)

例程说明：

STZRZF 例程通过正交变换将上梯形矩阵 A（M×N，M <=N）变换为上三角矩阵。上梯形矩阵 A 被分解为：

A = （R 0）*Z，

其中：Z 是 N×N 接正交矩阵，R 是 M×M 上三角矩阵。

参数说明：

M 输入参数，INTEGER

矩阵 A 的行数，M>=0。

N 输入参数，INTEGER

矩阵 A 的列数，N>=0。

A 输入输出参数，REAL 数组，dimension (LDA,N)

输入时，包含要分解的上梯形矩阵。

输出时，包含上三角矩阵，还有其余的 A 的元素。

LDA 输入参数，INTEGER

数组 A 的尺寸， LDA >= max(1,M)

TAU 输入参数，REAL 数组，dimension (M)

分解因子

INFO 输出参数，INTEGER

= 0：成功退出

<0：如果 INFO=-i,则第 i 号参数不被接受。

2.22 STRTRS 例程

原型：

SUBROUTINE STRTRS(UPLO, TRANS, DIAG, N, NRHS, A, LDA, B, LDB,INFO)

例程说明：

STRTRS 例程解决三角矩阵线性方程问题，A * X = B 或 A **T * X = B。A 是 N 维三角矩阵。B 是 N×NRHS 矩阵，A 是非奇异的。

矩阵 A 形式为：

A	0	0
0	B	
0		C

参数说明：

UPLO 输入参数，CHARACTER*1

= 'U'：A 是上三角矩阵

= 'L'：A 是下三角矩阵

TRANS 输入参数，CHARACTER*1

指定方程组的形式：

='N' A * X = B (不变换)

=C" A**H * X = B（变换）

='T' A**T * X = B （变换）

DIAG 输入参数，(input) CHARACTER*1

= 'N': A 为非单位三角矩阵

= 'U': A 为单位三角矩阵

N 输入参数，INTEGER

矩阵 A 的维数，N>=0

NRHS 输入参数，INTEGER

右边尺寸，如：B 矩阵的列数。NRHS>=0

A 输入矩阵，REAL 数组，dimension (LDA,N)

三角矩阵 A

LDA 输入参数，INTEGER

决定数组 A 的尺寸，LDA >= max(1,N)。

B 输入输出参数，REAL 数组，dimension (LDB,NRHS)

输入时，右边矩阵 B。

输出时，如果 INFO =0 为解矩阵 X。

LDB 输入参数，INTEGER

数组 B 的尺寸，LDB >= max(1,N)。

INFO 输出参数，INTEGER

= 0:成功退出，

< 0:如果 INFO =-i，第 i 个参数不可接受。

```
!    对STRTRS进行代码演示
     program STRTRS_test
         implicit none
         !DEC$ OBJCOMMENT LIB: 'lapack_win32.lib'
         !DEC$ OBJCOMMENT LIB: 'blas_win32.lib'
         CHARACTER*1 UPLO,TRANS,DIAG
         INTEGER N,NRHS,LDA,LDB,INFO
         REAL,ALLOCATABLE::A(:,:),B(:,:)
         UPLO = 'U'
         TRANS = 'N'
         DIAG = 'N'
         N =3
         NRHS = 1
         LDA = MAX(1,N)
         LDB = MAX(1,N)
         ALLOCATE(A(LDA,N),B(LDB,NRHS))
         A =RESHAPE((/1.1,0.0,0.0,&
                      0.0,4.5,0.0,&
                      0.0,0.0,2.7/),(/3,3/))
         B =RESHAPE((/1.1,1.5,1.6/),(/3,1/))
         WRITE(*,10) A
         CALL STRTRS( UPLO,TRANS,DIAG,N,NRHS,A,LDA,B,LDB,INFO)
         WRITE(*,*) '输出解：'
         WRITE(*,20) B
10       FORMAT(/4X,F,4X,F,4X,F)
20       FORMAT(/4X,F)
     end program STRTRS_test
```

```
"E:\FORTRAN\STRTRS_test\Debug\STRTRS_test.exe"

         1.1000000          0.0000000          0.0000000

         0.0000000          4.5000000          0.0000000

         0.0000000          0.0000000          2.7000000
 输出解：

         1.0000000

         0.3333333

         0.5925926
Press any key to continue
```

我们用 matlab 来验证一下：

```
>> A =[1.1,0.0,0.0;0.0,4.5,0.0;0.0,0.0,2.7];
>> B =[1.1;1.5;1.6];
>> X = A\B

X =

    1.0000
    0.3333
    0.5926
```

2.23 SPPTRI 例程

原型：

SUBROUTINE SPPTRI(UPLO, N, AP, INFO)

例程说明：

SPPTRI 通过 SPPTRF 返回的 Cholesky 分解形式的 A = U**T*U 或 A = L*L**T，计算出 real 型对称正定矩阵 A 的逆。

参数说明：

UPLO 输入参数，CHARACTER*1

= 'U': 上三角矩阵（因式）储存在 AP 中。

= 'L': 下三角矩阵（因式）储存在 AP 中。

N 输入参数，INTEGER

矩阵 A 的维数，N>=0。

AP 输入输出参数，REAL 数组，dimension (N*(N+1)/2)

输入时，AP 存储 A = U**T*U 或 A = L*L**T 中的 U 或 L.

AP 中 U 或 L 的第 j 列元素存储如下：

如果 UPLO = 'U'，则 AP(i + (j-1)*j/2) = U(i,j) ，1<=i<=j;

如果 UPLO = 'L'，则 AP(i + (j-1)*(2n-j)/2) = L(i,j)， j<=i<=n.

INFO 输出参数，INTEGER

=0：成功退出

<0:如果 INFO = -i，第 i 号元素不可接受

>0:如果 INFO=i，则 L 或 U 的（i，i）元素为 0，计算失败！

2.24 SPPTRF 例程

原型：

SUBROUTINE SPPTRF(UPLO, N, AP, INFO)

例程说明：

该例程通过 Cholesky 分解 REAL 型对称正定矩阵 A，并以紧凑格式存储。

参数说明:

UPLO 输入参数，CHARACTER*1

= 'U': A 的上三角矩阵被储存。

= 'L': A 的下三角矩阵被储存。

N 输入参数，INTEGER

矩阵 A 的维数，N>=0。

AP 输入输出参数，REAL 数组，dimension (N*(N+1)/2)

输入时，AP 存储 A = U**T*U 或 A = L*L**T 中的 U 或 L.

AP 中 U 或 L 的第 j 列元素存储如下：

如果 UPLO = 'U'，则 AP(i + (j-1)*j/2) = U(i,j) ，1<=i<=j;

如果 UPLO = 'L'，则 AP(i + (j-1)*(2n-j)/2) = L(i,j)， j<=i<=n.

输出时为 U 或 L。

INFO 输出参数，INTEGER

=0：成功退出

<0:如果 INFO = -i，第 i 号元素不可接受

>0:如果 INFO=i，则 L 或 U 的（i，i）元素为 0，计算失败！