



O USO DA BIBLIOTECA LAPACK

A **biblioteca LAPACK** – é um pacote de softwares mantido pelas Universidades do Tennessee, da Califórnia (Berkeley), do Colorado (Denver) e pela NAG Ltda. LAPACK significa *Linear Algebra PACKage*, ou, numa tradução livre, *Pacote de Álgebra Linear*.

A LAPACK é escrita em Fortran e fornece rotinas para solução de sistemas de equações lineares simultâneas, solução de sistemas de equações por mínimos quadrados, e para solução de problemas de autovalor. As matrizes associadas de fatorização (LU, Cholesky, QR, SVD, Schur e Schur generalizada) também são fornecidas, e alguns outros cálculos relacionados, tal como o reordenamento das fatorizações de Schur e os números condicionais estimados. As matrizes densas e em bandas também são tratadas pelos softwares, mas não as matrizes esparsas em geral. Em todos os itens citados acima, uma funcionalidade semelhante é fornecida para matrizes reais e complexas, em precisão simples e dupla.

Uso da biblioteca LAPACK – Para apresentar o uso da biblioteca LAPACK, faremos a solução de sistemas de equações lineares. Começaremos com o seguinte sistema de equações lineares:

$$\begin{cases} 2x_1 + x_2 = 1 \\ x_1 + 3x_2 = 2 \end{cases} \quad (1)$$

Mas antes, faça o cálculo em uma folha de papel. É fácil, e o resultado é $x_1 = 0,2$ e $x_2 = 0,6$. A solução de sistemas de equações lineares, pela LAPACK, é fornecida pela sub-rotina `SGESV`. O `S` no início do nome indica que a solução será com precisão simples. Caso seja necessário a solução em dupla precisão devemos usar a sub-rotina `DGESV`.

Acesse o site da LAPACK para *download* da biblioteca, e após identifique onde está o arquivo `sgesv.f` (sub-rotina `SGESV`) e todas as suas dependências. Esta sub-rotina foi desenvolvida com o FORTRAN 77, que é identificado pela extensão `.f`. Já está disponível a versão em Fortran 90. É interessante, ao visitar o site, que leia todas as instruções e notícias a respeito da biblioteca. Vale a pena “investir alguns minutos” para entender os objetivos da LAPACK.

A LAPACK está hospedada no site da NETLIB, cujo link é: <http://www.netlib.org> ou vá diretamente em <http://www.netlib.org/lapack/index.html>.

Antes de usarmos é necessário analisar e entender o que esta sub-rotina (`SGESV`) realiza, e verificar se ela utiliza (ou depende de) outras sub-rotinas e funções. A seguir é apresentada a “antiga” sub-rotina `SGESV`, na qual podemos ver diretamente que ela “chama” (`CALL`) outras três sub-rotinas, que são: `XERBLA`, `SGETRF` e `SGETRS`. É prudente verificar se essas sub-rotinas utilizam outras sub-rotinas. Neste caso, temos mais 14 sub-rotinas utilizadas pelas 3 sub-rotinas acima. Na versão atual, nos comentários da sub-rotina, são indicados *links* para baixar a sub-rotina e suas dependências.

Observe atentamente a descrição do uso da sub-rotina, que ocorre entre as linhas 16 e 65, com especial atenção para o detalhamento dos argumentos da sub-rotina, que inicia na linha 30. A sub-rotina possui 108 linhas, mas somente 27 linhas são de instruções “executáveis” da linguagem, as demais são informações importantes para o usuário da sub-rotina, como comentários e espaçamentos verticais para melhorar a visualização do código. Os comentários, no FORTRAN 77, são indicados pelo asterisco (*), que no Fortran 90 é o ponto de exclamação (!).

```
1      SUBROUTINE SGESV( N, NRHS, A, LDA, IPIV, B, LDB, INFO )
2      *
3      *  -- LAPACK driver routine (version 3.2) --
4      *  -- LAPACK is a software package provided by Univ. of Tennessee,  --
```

```

5 * -- Univ. of California Berkeley, Univ. of Colorado Denver and NAG Ltd.--
6 *   November 2006
7 *
8 *   .. Scalar Arguments ..
9 *   INTEGER          INFO, LDA, LDB, N, NRHS
10 *   ..
11 *   .. Array Arguments ..
12 *   INTEGER          IPIV( * )
13 *   REAL             A( LDA, * ), B( LDB, * )
14 *   ..
15 *
16 * Purpose
17 * =====
18 *
19 * SGESV computes the solution to a real system of linear equations
20 *   A * X = B,
21 * where A is an N-by-N matrix and X and B are N-by-NRHS matrices.
22 *
23 * The LU decomposition with partial pivoting and row interchanges is
24 * used to factor A as
25 *   A = P * L * U,
26 * where P is a permutation matrix, L is unit lower triangular, and U is
27 * upper triangular. The factored form of A is then used to solve the
28 * system of equations A * X = B.
29 *
30 * Arguments
31 * =====
32 *
33 * N          (input) INTEGER
34 *             The number of linear equations, i.e., the order of the
35 *             matrix A.  N >= 0.
36 *
37 * NRHS       (input) INTEGER
38 *             The number of right hand sides, i.e., the number of columns
39 *             of the matrix B.  NRHS >= 0.
40 *
41 * A          (input/output) REAL array, dimension (LDA,N)
42 *             On entry, the N-by-N coefficient matrix A.
43 *             On exit, the factors L and U from the factorization
44 *             A = P*L*U; the unit diagonal elements of L are not stored.
45 *
46 * LDA        (input) INTEGER
47 *             The leading dimension of the array A.  LDA >= max(1,N).
48 *
49 * IPIV        (output) INTEGER array, dimension (N)
50 *             The pivot indices that define the permutation matrix P;
51 *             row i of the matrix was interchanged with row IPIV(i).
52 *
53 * B          (input/output) REAL array, dimension (LDB,NRHS)
54 *             On entry, the N-by-NRHS matrix of right hand side matrix B.
55 *             On exit, if INFO = 0, the N-by-NRHS solution matrix X.
56 *
57 * LDB        (input) INTEGER
58 *             The leading dimension of the array B.  LDB >= max(1,N).
59 *
60 * INFO       (output) INTEGER
61 *             = 0:  successful exit

```

```

62 *          < 0:  if INFO = -i, the i-th argument had an illegal value
63 *          > 0:  if INFO = i, U(i,i) is exactly zero.  The factorization
64 *                has been completed, but the factor U is exactly
65 *                singular, so the solution could not be computed.
66 *
67 * =====
68 *
69 *      .. External Subroutines ..
70      EXTERNAL          SGETRF, SGETRS, XERBLA
71 *
72 *      .. Intrinsic Functions ..
73      INTRINSIC          MAX
74 *
75 *      .. Executable Statements ..
76 *
77 *      Test the input parameters.
78 *
79      INFO = 0
80      IF( N.LT.0 ) THEN
81          INFO = -1
82      ELSE IF( NRHS.LT.0 ) THEN
83          INFO = -2
84      ELSE IF( LDA.LT.MAX( 1, N ) ) THEN
85          INFO = -4
86      ELSE IF( LDB.LT.MAX( 1, N ) ) THEN
87          INFO = -7
88      END IF
89      IF( INFO.NE.0 ) THEN
90          CALL XERBLA( 'SGESV ', -INFO )
91          RETURN
92      END IF
93 *
94 *      Compute the LU factorization of A.
95 *
96      CALL SGETRF( N, N, A, LDA, IPIV, INFO )
97      IF( INFO.EQ.0 ) THEN
98 *
99 *          Solve the system A*X = B, overwriting B with X.
100 *
101          CALL SGETRS( 'No transpose', N, NRHS, A, LDA, IPIV, B, LDB,
102 $                  INFO )
103      END IF
104      RETURN
105 *
106 *      End of SGESV
107 *
108      END

```

A equação (1), acima, pode ser escrita como segue, em acordo com a linha 20 da sub-rotina SGESV,

$$AX = B,$$

ou, na forma matricial, já com pontos no lugar das vírgulas (porque no Fortran é usado ponto no lugar de vírgula para indicar a parte decimal),

$$\begin{pmatrix} 2.0 & 1.0 \\ 1.0 & 3.0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 2.0 \end{pmatrix}, \quad \text{em que} \quad A = \begin{pmatrix} 2.0 & 1.0 \\ 1.0 & 3.0 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad B = \begin{pmatrix} 1.0 \\ 2.0 \end{pmatrix}.$$

Escreva um programa que leia as duas matrizes acima (A e B) e use a sub-rotina *SGESV*, cuidando atentamente os seus parâmetros, para resolver o sistema de equações lineares e obter **X**, ou x_1 e x_2 . Compare esses resultados com os obtidos pelo cálculo feito no papel, e teste com outros sistemas de equações lineares. Faça o programa de tal forma que possa resolver qualquer sistema de equações, isto é, que não fique restrito a matriz 2x2.

Compilação da sub-rotina *SGESV* – Não é necessário agregar (escrever) a sub-rotina *SGESV* (e as suas dependências) diretamente no programa principal. As (17) sub-rotinas permanecem externas, mas no mesmo diretório do programa principal, e são chamadas somente na compilação, como indica o exemplo a seguir, para o caso do compilador no Windows (g95) e GNU/Linux (com o g95 instalado):

```
g95 -o SISTEMA sistemaeq.f90 ieeck.f ilaenv.f iparmq.f isamax.f lsame.f sgemm.f
sger.f sgesv.f sgetf2.f sgetrf.f sgetrs.f slamch.f slaswp.f sscal.f sswap.f strsm.f xerbla.f
```

e para o compilador no iMac (gfortran):

```
gfortran -o SISTEMA sistemaeq.f90 ieeck.f ilaenv.f iparmq.f isamax.f lsame.f
sgemm.f sger.f sgesv.f sgetf2.f sgetrf.f sgetrs.f slamch.f slaswp.f sscal.f sswap.f
strsm.f xerbla.f
```

em que **SISTEMA** é o nome do executável e o programa principal é *sistemaeq.f90*, e os demais são arquivos da LAPACK.

Se tivermos que compilar em diferentes dias ou momentos, devemos digitar, cada vez, toda a sequência acima. Para evitar esse procedimento, muitas vezes “*tedioso*”, escrevemos o conjunto de instruções em um arquivo do tipo *batch*. O *batch* ou *arquivo de lote* (também conhecido por **.bat**, no caso do Windows) é um arquivo de computador utilizado para automatizar tarefas. Assim, temos **NOME-ARQUIVO.BAT**, para o Windows, ou um arquivo **NOME-ARQUIVO.sh** em *shell script* para o iMac ou UNIX ou GNU/Linux, e o executamos quando necessário. Para o caso do iMac é importante, após gerar o arquivo, alterar a configuração de permissões do arquivo. Para isso, digite no terminal a instrução:

```
sudo chmod 700 nome-do-arquivo-script.sh
```

Exercício – Encontre a solução dos seguintes sistemas de equações lineares:

$$\begin{cases} 3x + 5y &= 1 \\ 2x &+ z = 3 \\ 5x + y - z &= 0 \end{cases}$$

(Solução: $x = \frac{7,0}{16,0} = 0,4375$, $y = -\frac{1,0}{16,0} = -0,0625$ e $z = \frac{17,0}{8,0} = 2,125$).

$$\begin{cases} x + 2y + z &= 12 \\ x - 3y + 5z &= 1 \\ 2x - y + 3z &= 10 \end{cases}$$

(Solução: $x = 5,0$, $y = 3,0$ e $z = 1,0$).

O código em Fortran para resolver o sistema de equações lineares – O seguinte programa recebe as matrizes do sistema de equações lineares e utiliza a sub-rotina *SGESV* para obter a solução.

```
1 PROGRAM testa_lapack
2 IMPLICIT NONE
3 ! Usa a sub-rotina SGESV da Lapack - solucao de sistema de equacoes lineares
```

```

4  ! www.netlib.org
5  ! Autor: Gilberto Orengo (g.orengo@gmail.com)
6  ! Obs.: os dados, das matrizes, sao oriundas de um Arquivo.
7  !       O nome do Arquivo eh fornecido pelo usuario
8  !
9  ! Declaracao das variaveis
10 REAL (KIND=4), ALLOCATABLE, DIMENSION(:, :) :: A, B, X
11 INTEGER (KIND=4), ALLOCATABLE :: IPIV(:)
12 INTEGER (KIND=4) :: N, NRHS, LDA, LDB, INFO, i, j
13 CHARACTER (LEN=20) :: nome
14 !
15 WRITE(*,*) "O nome do arquivo com os dados (inclua a extensao): "
16 READ(*,*) nome
17 !
18 OPEN(UNIT=30, FILE=nome)
19 READ(30,*) N
20 READ(30,*) NRHS
21 LDA=N
22 LDB=N
23 ALLOCATE(A(LDA,N))      ! Alocacao dinamica de memoria da matriz A
24 ALLOCATE(B(LDB,NRHS))   ! Alocacao dinamica de memoria da matriz B
25 ALLOCATE(IPIV(N))       ! Alocacao dinamica de memoria do vetor IPIV
26 !
27 ! Formatacao de saida
28 10 FORMAT(A,I2,A,I2,A)
29 20 FORMAT(A,I2,A,I2,A,F8.5)
30 !
31 WRITE(*,*) "Matriz A:" ! Le e Escreve a matriz A, para conferencia
32 DO i=1,LDA
33     DO j=1,N
34         READ(30,*) A(i,j)
35         WRITE(*,20) "A(",i," ",j,"): ",A(i,j)
36     END DO
37 END DO
38 !
39 WRITE(*,*) "Matriz B:" ! Le e Escreve a matriz B, para conferencia
40 DO i=1,LDB
41     DO j=1,NRHS
42         READ(30,*) B(i,j)
43         WRITE(*,20) "B(",i," ",j,"): ",B(i,j)
44     END DO
45 END DO
46 !
47 ! Chamada da sub-rotina da LAPACK
48 CALL SGESV(N,NRHS,A,LDA,IPIV,B,LDB,INFO)
49 X=B ! A matriz B ao sair da sub-rotina traz o resultado X
50 WRITE(*,*) " "
51 WRITE(*,*) "RESULTADO:" ! Escreve o resultado (X)
52 DO i=1,LDB
53     DO j=1,NRHS
54         WRITE(*,20) "X(",i," ",j,"): ",X(i,j)
55     END DO
56 END DO
57 WRITE(*,*) " "
58 CLOSE(30)
59 END PROGRAM testa_lapack

```