NAG Toolbox

nag_lapack_zgels (f08an)

1 Purpose

nag_lapack_zgels (f08an) solves linear least squares problems of the form

$$\min_x \|b - Ax\|_2 \quad \text{or} \quad \min_x \|b - A^{\mathrm{H}}x\|_2,$$

where A is an m by n complex matrix of full rank, using a QR or LQ factorization of A.

2 Syntax

```
[a, b, info] = nag_lapack_zgels(trans, a, b, 'm', m, 'n', n, 'nrhs_p', nrhs_p)
[a, b, info] = f08an(trans, a, b, 'm', m, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

The following options are provided:

1. If **trans** = 'N' and $m \geq n$: find the least squares solution of an overdetermined system, i.e., solve the least squares problem

$$\min_x \|b - Ax\|_2.$$

2. If **trans** = 'N' and $m < n$: find the minimum norm solution of an underdetermined system $Ax = b$.
3. If **trans** = 'C' and $m \geq n$: find the minimum norm solution of an undetermined system $A^{\mathrm{H}}x = b$.
4. If **trans** = 'C' and $m < n$: find the least squares solution of an overdetermined system, i.e., solve the least squares problem

$$\min_x \|b - A^{\mathrm{H}}x\|_2.$$

Several right-hand side vectors b and solution vectors x can be handled in a single call; they are stored as the columns of the m by r right-hand side matrix B and the n by r solution matrix X.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

5.1 Compulsory Input Parameters

1: **trans** – CHARACTER(1)

If **trans** = 'N', the linear system involves A.

If **trans** = 'C', the linear system involves A^{H}.

Constraint: **trans** = 'N' or 'C'.

2: **a**$(lda,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** must be at least $\max(1,\mathbf{m})$.

The second dimension of the array **a** must be at least $\max(1,\mathbf{n})$.

The m by n matrix A.

3: **b**$(ldb,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **b** must be at least $\max(1,\mathbf{m},\mathbf{n})$.

The second dimension of the array **b** must be at least $\max(1,\mathbf{nrhs\_p})$.

The matrix B of right-hand side vectors, stored in columns; **b** is m by r if **trans** = 'N', or n by r if **trans** = 'C'.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the first dimension of the array **a**.

m, the number of rows of the matrix A.

Constraint: $\mathbf{m} \geq 0$.

2: **n** – INTEGER

Default: the second dimension of the array **a**.

n, the number of columns of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

3: **nrhs_p** – INTEGER

Default: the second dimension of the array **b**.

r, the number of right-hand sides, i.e., the number of columns of the matrices B and X.

Constraint: $\mathbf{nrhs\_p} \geq 0$.

5.3 Output Parameters

1: **a**$(lda,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1,\mathbf{m})$.

The second dimension of the array **a** will be $\max(1,\mathbf{n})$.

If $\mathbf{m} \geq \mathbf{n}$, **a** stores details of its QR factorization, as returned by nag_lapack_zgeqrf (f08as).

If $\mathbf{m} < \mathbf{n}$, **a** stores details of its LQ factorization, as returned by nag_lapack_zgelqf (f08av).

2: **b**$(ldb,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **b** will be $\max(1,\mathbf{m},\mathbf{n})$.

The second dimension of the array **b** will be $\max(1,\mathbf{nrhs\_p})$.

b stores the solution vectors, x, stored in columns:

if **trans** = 'N' and $m \geq n$, or **trans** = 'C' and $m < n$, elements 1 to $\min(m,n)$ in each column of **b** contain the least squares solution vectors; the residual sum of squares for the solution is given by the sum of squares of the modulus of elements $(\min(m,n)+1)$ to $\max(m,n)$ in that column;

otherwise, elements 1 to $\max(m,n)$ in each column of **b** contain the minimum norm solution vectors.

3: **info** – INTEGER

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **trans**, 2: **m**, 3: **n**, 4: **nrhs_p**, 5: **a**, 6: **lda**, 7: **b**, 8: **ldb**, 9: **work**, 10: **lwork**, 11: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

info > 0

If **info** $= i$, diagonal element i of the triangular factor of A is zero, so that A does not have full rank; the least squares solution could not be computed.

7 Accuracy

See Section 4.5 of Anderson *et al.* (1999) for details of error bounds.

8 Further Comments

The total number of floating-point operations required to factorize A is approximately $\frac{8}{3}n^2(3m-n)$ if $m \geq n$ and $\frac{8}{3}m^2(3n-m)$ otherwise. Following the factorization the solution for a single vector x requires $O\left(\min\left(m^2, n^2\right)\right)$ operations.

The real analogue of this function is nag_lapack_dgels (f08aa).

9 Example

This example solves the linear least squares problem

$$\min_x \|b - Ax\|_2,$$

where

$$A = \begin{pmatrix} 0.96 - 0.81i & -0.03 + 0.96i & -0.91 + 2.06i & -0.05 + 0.41i \\ -0.98 + 1.98i & -1.20 + 0.19i & -0.66 + 0.42i & -0.81 + 0.56i \\ 0.62 - 0.46i & 1.01 + 0.02i & 0.63 - 0.17i & -1.11 + 0.60i \\ -0.37 + 0.38i & 0.19 - 0.54i & -0.98 - 0.36i & 0.22 - 0.20i \\ 0.83 + 0.51i & 0.20 + 0.01i & -0.17 - 0.46i & 1.47 + 1.59i \\ 1.08 - 0.28i & 0.20 - 0.12i & -0.07 + 1.23i & 0.26 + 0.26i \end{pmatrix}$$

and

$$b = \begin{pmatrix} -2.09 + 1.93i \\ 3.34 - 3.53i \\ -4.94 - 2.04i \\ 0.17 + 4.23i \\ -5.19 + 3.63i \\ 0.98 + 2.53i \end{pmatrix}.$$

The square root of the residual sum of squares is also output.

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

9.1 Program Text

```
function f08an_example

fprintf('f08an example results\n\n');

a = [ 0.96 - 0.81i, -0.03 + 0.96i,  -0.91 + 2.06i, -0.05 + 0.41i;
     -0.98 + 1.98i, -1.20 + 0.19i,  -0.66 + 0.42i, -0.81 + 0.56i;
      0.62 - 0.46i,  1.01 + 0.02i,   0.63 - 0.17i, -1.11 + 0.60i;
     -0.37 + 0.38i,  0.19 - 0.54i,  -0.98 - 0.36i,  0.22 - 0.20i;
      0.83 + 0.51i,  0.20 + 0.01i,  -0.17 - 0.46i,  1.47 + 1.59i;
      1.08 - 0.28i,  0.20 - 0.12i,  -0.07 + 1.23i,  0.26 + 0.26i];
b = [-2.09 + 1.93i;
      3.34 - 3.53i;
     -4.94 - 2.04i;
      0.17 + 4.23i;
     -5.19 + 3.63i;
      0.98 + 2.53i];
[m,n] = size(a);

% Solve the least squares problem min( norm2(b - Ax) ) for x
trans = 'No transpose';
[af, x, info] = f08an( ...
        trans, a, b);

% Print Solution
fprintf('\nLeast Squares Solution:\n');
disp(transpose(x(1:n)));
fprintf('Square root of the residual sum of squares\n');
disp(norm(x(n+1:m),2));
```

9.2 Program Results

```
f08an example results

Least Squares Solution:
  -0.5044 - 1.2179i  -2.4281 + 2.8574i   1.4872 - 2.1955i   0.4537 + 2.6904i

Square root of the residual sum of squares
    0.0688
```