

NAG Toolbox

nag_lapack_zgelsd (f08kq)

1 Purpose

`nag_lapack_zgelsd (f08kq)` computes the minimum norm solution to a complex linear least squares problem

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

2 Syntax

```
[a, b, s, rank, info] = nag_lapack_zgelsd(a, b, rcond, 'm', m, 'n', n, 'nrhs_p',
nrhs_p, 'lwork', lwork)
[a, b, s, rank, info] = f08kq(a, b, rcond, 'm', m, 'n', n, 'nrhs_p', nrhs_p,
'lwork', lwork)
```

3 Description

`nag_lapack_zgelsd (f08kq)` uses the singular value decomposition (SVD) of A , where A is a complex m by n matrix which may be rank-deficient.

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 .

The problem is solved in three steps:

1. reduce the coefficient matrix A to bidiagonal form with Householder transformations, reducing the original problem into a ‘bidiagonal least squares problem’ (BLS);
2. solve the BLS using a divide-and-conquer approach;
3. apply back all the Householder transformations to solve the original least squares problem.

The effective rank of A is determined by treating as zero those singular values which are less than **rcond** times the largest singular value.

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: **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 coefficient matrix A .

2: **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 m by r right-hand side matrix B .

3: **rcond** – REAL (KIND=nag_wp)

Used to determine the effective rank of A . Singular values $s(i) \leq \mathbf{rcond} \times s(1)$ are treated as zero. If $\mathbf{rcond} < 0$, **machine precision** is used instead.

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: $\mathbf{nrhs_p} \geq 0$.

4: **lwork** – INTEGER

Suggested value: for optimal performance, **lwork** should generally be larger than the required minimum. Consider increasing **lwork** by at least $nb \times \min(\mathbf{m}, \mathbf{n})$, where nb is the optimal **block size**.

Default: $\max(1, 64 \min(\mathbf{m}, \mathbf{n}) \max(\mathbf{m} + \mathbf{n} + r, 2r + r \times \mathbf{nrhs_p}))$

The dimension of the array **work**.

The exact minimum amount of workspace needed depends on **m**, **n** and **nrhs_p**. As long as **lwork** is at least

$$\max(1, \mathbf{m} + \mathbf{n} + r, 2r + r \times \mathbf{nrhs_p}),$$

where $r = \min(\mathbf{m}, \mathbf{n})$, the code will execute correctly.

Constraint: **lwork** must be at least $\max(1, \mathbf{m} + \mathbf{n} + r, 2r + r \times \mathbf{nrhs_p})$.

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 $\max(1, \mathbf{n})$.

The contents of **a** are destroyed.

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 n by r solution matrix X . If $m \geq n$ and **rank** = n , the residual sum of squares for the solution in the i th column is given by the sum of squares of the modulus of elements $n+1, \dots, m$ in that column.

3: **s**(:) – REAL (KIND=nag_wp) array

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

The singular values of A in decreasing order.

4: **rank** – INTEGER

The effective rank of A , i.e., the number of singular values which are greater than **rcond** \times **s**(1).

5: **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: **m**, 2: **n**, 3: **nrhs_p**, 4: **a**, 5: **ida**, 6: **b**, 7: **ldb**, 8: **s**, 9: **rcond**, 10: **rank**, 11: **work**, 12: **lwork**, 13: **rwork**, 14: **iwork**, 15: **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

The algorithm for computing the SVD failed to converge; if **info** = i , i off-diagonal elements of an intermediate bidiagonal form did not converge to zero.

7 Accuracy

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

8 Further Comments

The real analogue of this function is nag_lapack_dgelsd (f08kc).

9 Example

This example solves the linear least squares problem

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

for the solution, x , of minimum norm, where

$$A = \begin{pmatrix} 0.47 - 0.34i & -0.32 - 0.23i & 0.35 - 0.60i & 0.89 + 0.71i & -0.19 + 0.06i \\ -0.40 + 0.54i & -0.05 + 0.20i & -0.52 - 0.34i & -0.45 - 0.45i & 0.11 - 0.85i \\ 0.60 + 0.01i & -0.26 - 0.44i & 0.87 - 0.11i & -0.02 - 0.57i & 1.44 + 0.80i \\ 0.80 - 1.02i & -0.43 + 0.17i & -0.34 - 0.09i & 1.14 - 0.78i & 0.07 + 1.14i \end{pmatrix}$$

and

$$b = \begin{pmatrix} 2.15 - 0.20i \\ -2.24 + 1.82i \\ 4.45 - 4.28i \\ 5.70 - 6.25i \end{pmatrix}.$$

A tolerance of 0.01 is used to determine the effective rank of A .

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 f08kq_example

fprintf('f08kq example results\n\n');

% Least squares problem min ||b - Ax|| where A and b are:
a = [ 0.47 - 0.34i, -0.32 - 0.23i, 0.35 - 0.60i, 0.89 + 0.71i, -0.19 + 0.06i;
      -0.40 + 0.54i, -0.05 + 0.20i, -0.52 - 0.34i, -0.45 - 0.45i, 0.11 - 0.85i;
      0.60 + 0.01i, -0.26 - 0.44i, 0.87 - 0.11i, -0.02 - 0.57i, 1.44 + 0.80i;
      0.80 - 1.02i, -0.43 + 0.17i, -0.34 - 0.09i, 1.14 - 0.78i, 0.07 + 1.14i];
[m,n] = size(a);
b = [ 2.15 - 0.20i;
      -2.24 + 1.82i;
      4.45 - 4.28i;
      5.70 - 6.25i;
      0     + 0i];

% Treat singular values less than 0.01 as zero
rcond = 0.01;
[vr, x, s, rank, info] = f08kq( ...
                           a, b, rcond);

disp('Least squares solution');
disp(x(1:n));
disp('Tolerance used to estimate the rank of A');
fprintf('%12.2e\n',rcond);
disp('Estimated rank of A');
fprintf('%5d\n',rank);
disp('Singular values of A');
disp(s');
```

9.2 Program Results

```
f08kq example results

Least squares solution
 3.9747 - 1.8377i
 -0.9186 + 0.8253i
 -0.3105 + 0.1477i
  1.0050 + 0.8626i
 -0.2256 - 1.9425i

Tolerance used to estimate the rank of A
 1.00e-02
Estimated rank of A
 3

Singular values of A
 2.9979    1.9983    1.0044    0.0064
```
