hqr

Purpose

Hyperbolic QR factorization

Syntax

```
[Q,R,Z] = hqr(A,J)

[Q,R,Z] = hqr(A,J,tol)
```

Description

Given a square matrix A and a diagonal signature matrix J, the function computes a hyperbolic QR factorization of A such that matrix

$$QA = R$$

has as many zero rows as the rank defect of A, and matrix Q is J-orthogonal, i.e.

$$Q^T JQ = J.$$

Output vector `Z` contains indices of non-zero rows in matrix R.

The above factorization exists if and only if

$$\operatorname{rank} A = \operatorname{rank} A^T JA$$

otherwise the function returns empty output arguments.

An optional tolerance parameter `tol` can be specified as an input argument. It is used as an absolute threshold to detect zero elements during the reduction process. By default it is the global zeroing tolerance.

Examples

Consider first a standard example. One can check that the obtained matrix `Q` is not orthogonal.

```
>> A
A =
```

```
     1     1     1     0
     0     0     1     0
     0     0     1     0
     0     0     1     0

>> J
J =

    -1     0     0     0
     0     1     0     0
     0     0    -1     0
     0     0     0     1

>> [Q,R,Z] = hqr(A,J)
Q =
    1.0000         0         0         0
         0    1.0000   -1.0000    1.0000
         0   -0.7071    1.4142   -0.7071
         0   -0.7071         0    0.7071
R =
    1.0000    1.0000    1.0000         0
         0         0    1.0000         0
         0         0   -0.0000         0
         0         0         0         0
Z =
     1     2

>> svd(Q)
ans =
    2.4142
    1.0000
    1.0000
    0.4142
```

This second example shows that the location of zero rows in R is not arbitrary.

```
>> A
A =
     1     0     1     8
     0     5     0     0
     1     2     1     8
     7     8     7     8

>> J
J =
```

```
     1     0     0     0
     0     1     0     0
     0     0    -1     0
     0     0     0    -1

>> [Q,R,Z] = hqr(A,J)
Q =
    2.5000    1.0000   -2.5000   -0.0000
   -0.2337    0.9872   -0.1612    0.0564
   -2.2987   -0.9872    2.6936   -0.0564
   -0.1429         0    0.1429    1.0000
R =
   -0.0000         0   -0.0000    0.0000
   -0.0000    5.0649   -0.0000   -2.7077
    0.0000         0    0.0000    2.7077
    7.0000    8.2857    7.0000    8.0000
Z =
     4     2     3
```

Finally, this example shows that when the rank assumption on input matrices `A` and `J` is not satisfied, then function `hqr` returns empty output arguments.

```
>> A
A =
     1     0
     1     0

>> J
J =
     1     0
     0    -1

>> rank(A'*J*A)
ans =
     0

>> [Q,R,Z]=hqr(A,J)
Q =
     []
R =
     []
Z =
     []
```

Algorithm

The algorithm is described in reference [1]. It consists in zeroing iteratively entries in each column of A by application of orthogonal (plane) rotations, row permutations and J-orthogonal (hyperbolic) rotations. Note that, unlike orthogonal rotations, J-orthogonal rotations may be potentially numerically instable.

References

[1] D. Henrion, P. Hippe. Hyperbolic QR factorization for J-spectral factorization of polynomial matrices. LAAS-CNRS Research Report, Toulouse, France, February 2003

Diagnostics

The function issues error messages under the following conditions:

- Input matrix `A` is not square.
- Input matrix `J` has a zero diagonal entry.

See also

`qr` Standard QR factorization.