

```

function [Q,R] = buildhouseQR(A)
%
%   Computes the QR decomposition of a square
%   matrix A (possibly singular) using Householder reflections

n = size(A,1);
[R,u] = houseqr(A);
Q = buildQ(u);
% Produces a matrix R whose diagonal entries are
% nonnegative
P = eye(n);
for i = 1:n
    if R(i,i) < 0
        P(i,i) = -1;
    end
end
Q = Q*P; R = P*R;
end

```

Example 13.1. Consider the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{pmatrix}.$$

Running the function `buildhouseQR`, we get

$$Q = \begin{pmatrix} 0.1826 & 0.8165 & 0.4001 & 0.3741 \\ 0.3651 & 0.4082 & -0.2546 & -0.7970 \\ 0.5477 & -0.0000 & -0.6910 & 0.4717 \\ 0.7303 & -0.4082 & 0.5455 & -0.0488 \end{pmatrix}$$

and

$$R = \begin{pmatrix} 5.4772 & 7.3030 & 9.1287 & 10.9545 \\ 0 & 0.8165 & 1.6330 & 2.4495 \\ 0 & -0.0000 & 0.0000 & 0.0000 \\ 0 & -0.0000 & 0 & 0.0000 \end{pmatrix}.$$

Observe that A has rank 2. The reader should check that $A = QR$.

Remark: Curiously, running Matlab built-in function `qr`, the same R is obtained (up to column signs) but a different Q is obtained (the last two columns are different).