The following result can be shown.

Theorem 18.2. *Every $n \times n$ complex or real matrix A is similar to an upper Hessenberg matrix H, that is, $A = UHU^*$ for some unitary matrix U. Furthermore, U can be constructed as a product of Householder matrices (the definition is the same as in Section 13.1, except that W is a complex vector, and that the inner product is the Hermitian inner product on $\mathbb{C}^n$). If A is a real matrix, then U is an orthogonal matrix (and H is a real matrix).*

Theorem 18.2 and algorithms for converting a matrix to Hessenberg form are discussed in Trefethen and Bau [176] (Lecture 26), Demmel [48] (Section 4.4.6, in the real case), Serre [156] (Theorem 13.1), and Meyer [125] (Example 5.7.4, in the real case). The proof of correctness is not difficult and will be the object of a homework problem.

The following functions written in `Matlab` implement a function to compute a Hessenberg form of a matrix.

The function `house` constructs the normalized vector u defining the Householder reflection that zeros all but the first entries in a vector x.

```
function [uu, u] = house(x)
tol = 2*10^(-15);   % tolerance
uu = x;
p = size(x,1);
% computes l^1-norm of x(2:p,1)
n1 = sum(abs(x(2:p,1)));
if n1 <= tol
   u = zeros(p,1);  uu = u;
else
   l = sqrt(x'*x);  % l^2 norm of x
   uu(1) = x(1) + signe(x(1))*l;
   u = uu/sqrt(uu'*uu);
end
end
```

The function `signe(z)` returms -1 if $z < 0$, else $+1$.

The function `buildhouse` builds a Householder reflection from a vector uu.

```
function P = buildhouse(v,i)
% This function builds a Householder reflection
%   [I 0 ]
%   [0 PP]
%   from a Householder reflection
%   PP = I - 2uu*uu'
%   where uu = v(i:n)
```