function [x,U,D] = bierman(z,R,H,xin,Uin,Din)   
%   
% Matlab implementation of the   
% Bierman ``square root filtering without square roots''   
%   
% From the diskette included with   
% M. S. Grewal, L. R. Weill and A. P. Andrews   
% Global Positioning Systems, Inertial Navigation and Integration   
% John Wiley & Sons, 2000.   
%   
% INPUTS:   
% z measurement (SCALAR)   
% R variance of measurement error   
% H measurement sensitivity (row) vector   
% xin a priori estimate of state vector   
% Uin unit upper triangular factor of covariance matrix of a priori state uncertainty   
% Din diagonal factor of covariance matrix of a priori state uncertainty   
% OUTPUTS:   
% x a posteriori estimate of state vector   
% U upper triangular UD factor of a posteriori state uncertainty covariance   
% D diagonal UD factor of a posteriori state uncertainty covariance   
%   
x = xin; % Store inputs into outputs,   
U = Uin; % because algorithm does in-place   
D = Din; % (destructive) calculation of outputs.   
a = U'\*H'; % a is not modified, but   
b = D\*a; % b is modified to become unscaled Kalman gain.   
dz = z - H\*xin;   
alpha = R;   
gamma = 1/alpha;   
 for j=1:length(xin),   
 beta = alpha;   
 alpha = alpha + a(j)\*b(j);   
 lambda = -a(j)\*gamma;   
 gamma = 1/alpha;   
 D(j,j) = beta\*gamma\*D(j,j);   
 for i=1:j-1,   
 beta = U(i,j);   
 U(i,j) = beta + b(i)\*lambda;   
 b(i) = b(i) + b(j)\*beta;   
 end;   
 end   
dzs = gamma\*dz; % apply scaling to innovations   
x = x + dzs\*b; % multiply by unscaled Kalman gain   
return;