```
% b)

function [X] = SimSDEJacobi( V0, X0, kappa, theta, ...
    sigma, r, rho, T, v_min, v_max, NumSim, NT)
% Simulation of Jacobi stochastic volatility model via
%   Euler discretization scheme

% V0,X0: parameters of the basis vector B for the first Hermite moment
% kappa, sigma, theta, r, rho: parameters of the Jacobi model
% T: maturity time (starting from zero t=0)
% v_min, v_max: parameters of the quadratic form
% NumSim: number of simulations
% NT: number of time intervals
% X: simulated points at final time

dt = T / NT;    % time step

% function Q
Q = @(v) (v-v_min).*(v_max-v)/(sqrt(v_max)-sqrt(v_min))^2;

% paths initialization
X = X0 * ones(NumSim,1);
V = V0 * ones(NumSim,1);

% time loop
for t = 1:NT
    dW1 = sqrt(dt) * randn(NumSim,1);
    dW2 = sqrt(dt) * randn(NumSim,1);

    V = V + kappa * (theta-V) * dt + sigma * sqrt(max(0,Q(V))) .* dW1;
    X = X + (r - 0.5 * V) * dt + rho * sqrt(max(0,Q(V))) .* dW1 + ...
        sqrt(max(0,V - rho^2 * Q(V))) .* dW2;
end

end
```