```
% d)

function [price] = PriceApprox(N, V0, X0, kappa, sigma, theta, r, ...
    rho, T, v_min, v_max, mu_w, sigma_w, k)
% Compute the approximation of the European call price in the Jacobi↙
model

% N: number of first coefficients to compute
% 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
% mu_w, sigma_w: mean ans standard deviation of the w(x) gaussian↙
density
% k: strike

% price: approximation of the European call price

l = HermiteMoments (N, V0, X0, kappa, sigma, theta, r, rho, T, ...
    v_min, v_max, mu_w, sigma_w);

price = zeros(length(k),1);     % vector initialization

for i = 1:length(k)

    % Fourier coefficients for European call option
    f = FourierCoefficients(N, r, T, k(i), mu_w, sigma_w);

    price(i) = dot(l,f);        % option price as scalar product
end

end
```