% Read First: Introduction

%This Matlab code implements the recursive procedure to compute
%the Laplace transform of the transition density function
%for diffusion processes developed in

%Goovaerts, Marc J., Roger J. A. Laeven and Zhaoning Shang (2012).
%Transform Analysis and Asset Pricing for Diffusion Processes: A Recursive Approach,
%The Journal of Computational Finance 16, 47-81,
%http://www.rogerlaeven.com/.

%To use:
%[1.] Specify the partition and the Laplace transform parameter.
%[2.] Specify the potential V(x).

%Copyright © 2012, Marc J. Goovaerts, Roger J. A. Laeven and Zhaoning Shang.

%This code may be redistributed and used freely provided that the following conditions are met:
%(i) Reference should be made to Goovaerts, Laeven and Shang (2012).
%(ii) Redistributions must retain the above copyright notice, this list of conditions and the
%following disclaimer.

%DISCLAIMER: This software is provided by the copyright holders "as is" and
%any express or implied warranties, including, but not
%limited to, the implied warranties of merchantability and
%fitness for a particular purpose, are disclaimed. In no event
%shall the copyright owners be liable for any direct,
%indirect, incidental, special, exemplary, or consequential
%damages (including, but not limited to, procurement of
%substitute goods or services; loss of use, data, or profits;
%or business interruption) however caused and on any theory
%of liability, whether in contract, strict liability, or
%tort (including negligence or otherwise) arising in any
%way out of the use of this software, even if advised of
%the possibility of such damage.

%Comments, suggestions and bug reports are welcome and should be addressed to Roger J. A. Laeven.
%% Main Program

% Recursive computation of the
% Laplace transform of the transition density function
% for a special case of the Vasicek (1977) model: potential V(x) = x^2.
% See Section 4.1 of Goovaerts, Laeven and Shang (2012) for further details.

clear;
tic;

% [1.] Specify the partition and the Laplace transform parameter
s = 0.1; % Laplace transform parameter
a0 = -10; % Lower limit of the partition
N = 400; % Number of recursions, has to be even

% Partition
da = 2*-a0/N; % Width of the partition
x = a0:da:-a0; % Partition

% Create a meshgrid for computing rho
X = repmat(x,N+1,1);
Y = repmat(x',1,N+1);

% Compute rho on the meshgrid
s2 = sqrt(2*s);
rho = exp(-s2*abs(X-Y))/s2; % Initial value of the recursion

% [2.] Specify the potential V(x)
V = x.^2;

% Partition width times potential V(x)
da2 = da*V;

% Start recursion
for m = 1:N+1
    rho = rho-rho(:,m)*rho(m,:)*da2(m)/(1+rho(m,m)*da2(m));
end
% Produce results
computing_time = toc
rho = max(rho, 0);
surf(x, x, rho); view(-10, 30); zlim([0 max(max(rho))]); shading flat;
title('Laplace transform of the transition density\newline(potential \(V(x)=x^2\); computed by recursion')
xlabel('\(x_0\)'), ylabel('x'), zlabel('\(\rho^s(x, x_0)\)');

%%% End