Appendix: A MATLAB program that illustrates the LQ method

```
% This program computes the value function and the optimal
% decision rules of the linear-quadratic approximation to the
% problem described in Exercise 6.

% Author:  Jorge Duran
% e-mail:  xurxo@eco.uc3m.es

% This message is for the loop below.

disp(' The program is iterating on Bellman's equation.  Please wait...')

% Step 0:  Input the parameter values

a = 0.33; % alpha
b = 0.96; % beta
p = 0.95; % rho
d = 0.10; % delta
e = exp(1); % Number e (intrascendent but necessary)

% Step 1:  Compute the steady state

K = ((a*b)/(1-b*(1-d)))^(1/(1-a));
X = d*K;
Z = 0;

% Step 2:  Construct the quadratic expansion of the utility function

% Step 2.1:  Evaluate all the first and second order derivatives
% of the utility function at the steady state:

R = log(e^Z*K^a - X);

Jz = (e^Z*K^a) / (e^Z*K^a - X);
Jk = (e^Z*a*K^(a-1)) / (e^Z*K^a - X);
Jx = (-1) / (e^Z*K^a - X);

Hzz = (((e^Z*K^a) * (e^Z*K^a - X)) - (e^Z*K^a)^2) / ((e^Z*K^a - X)^2);
Hkk = (((e^Z*a*(a-1)*K^(a-2))*(e^Z*K^a - X)) - ...
(e^Z*a*K^(a-1))^2 )/((e^Z* K^a - X)^2);
Hxx = (-1)/((e^Z*K^a - X)^2);
Hzk = (((e^Z*a*K^(a-1))*(e^Z*K^a - X)) - (e^Z*K^a*e^Z*a*K^(a-1)))/ ...
((e^Z*K^a - X)^2);
Hzx = (e^Z*K^a) / ((e^Z*K^a - X)^2);
Hkx = (e^Z*a*K^(a-1))/((e^Z*K^a - X)^2);
```

```
% Step 2.2:  Define the Jacobian vector and the Hessian matrix evaluated
% at the steady state

DJ = [Jz Jk Jx ]';
D2H = [Hzz Hzk Hzx
Hzk Hkk Hkx
Hzx Hkx Hxx];

% Step 2.3:  Define matrix Q

W = [Z K X]';

Q11 = R - W'*DJ + 0.5*W'*D2H*W;
Q12 = 0.5*(DJ-D2H*W);
Q22 = 0.5*D2H;

Q=[ Q11 Q12'
Q12 Q22];

% Step 3:  Compute the optimal value function.

% Step 3.1:  Partition matrix Q to separate the state and control variables

Qff = Q(1:3,1:3);
Qfd = Q(4,1:3);
Qdd = Q(4,4);

% Step 3.2:  Input matrix B

B = [ 1 0 0 0
0 p 0 0
0 0 1-d 1];

% Step 3.3:  Input matrix P0

P = [-0.1 0 0
0 -0.1 0
0 0 -0.1 ];

% Step 3.4:  Initialize auxiliary matrix A

A=ones(3);

% Step 3.5:  Iterate on Bellman's equation until convergence

while (norm(A-P)/norm(A))>0.0000001
A = P;
M=B'*P*B;
```

```
Mff = M(1:3,1:3);
Mfd = M(4,1:3);
Mdd = M(4,4);
P=Qff + (b*Mff) - (Qfd'+(b*Mfd)')*inv(Qdd +(b*Mdd))*(Qfd+(b*Mfd));
end

% Step 4:  Output the results

disp(' ')
disp(' The linear policy for investment is d*=J'F, where vector J is:')
J=-(inv(Qdd+(b*Mdd))*(Qfd+(b*Mfd)))';
J
disp(' ')
disp(' The optimal value function is V*=F'PF, where matrix P is:')
P

% END OF MATLAB FILE
```

The output of this program should be the following:

```
J =

0.4983
0.8607
-0.0411

P =

-0.4025 8.0839 0.7369
8.0839 1.0029 -0.1915
0.7369 -0.1915 -0.0819
```