Question: 4 Write down the Euler-Lagrange equations (KKT conditions) for this problem and verify they are:

$$\begin{aligned} x_{i+1} &= ax_i + bu_i \qquad x_0 = \underline{x}_0 \\ \lambda_i &= qx_i + a\lambda_{i+1} \qquad \lambda_N = px_N \\ 0 &= ru_i + b\lambda_{i+1} \end{aligned}$$

□

Question: 5 Solve the stationarity condition with respect to u_i, i.e. express u_i as function of λ_{i+1}. □

Question: 6 Reverse the recursion for the costate, i.e. express λ_{i+1} as function of x_i and λ_i. □

Question: 7 Assume now, that both the initial x_0 and λ_0 are given. Verify (check, convince yourself or simply accept) that the Euler-Lagrange equations are equivalent to

$$\begin{aligned} \lambda_{i+1} &= \frac{\lambda_i - qx_i}{a} \\ u_i &= -\frac{b}{r}\lambda_{i+1} \\ x_{i+1} &= ax_i + bu_i \end{aligned} \tag{1}$$

which can be solved for $i = 0,\ 1,\ \ldots\ N-1$. □

The problem in the method described above is that the initial value λ_0 is not known. However the costate at the end is known to obey (the end point constraint):

$$\lambda_N = px_N \quad \text{or} \quad \lambda_N - px_N = 0$$

One method is to guess λ_0 and check if the end point condition on the costate is satisfied. Notice, x_0 is known (given by the problem).

The solution to the following question can be found in fejlf.m.

Question: 8 Write a piece of matlab code that solves the recursions in (1). The input to the function is a guess on the initial costate (ie. λ_0) and the output is the error between the final costate and is correct value. □

The solution to the next two questions can be found in runex3.m.

Question: 9 Use eg. fsolve (in matlab) to find the correct initial costate value. □

Question: 10 plot the variation of x_i and u_i. Study the effect of the parameters p, r and q by changing their values. Try eg. $r = 10q$ and $r = 0.1q$ and $p = 0$ and $p = 100 * q$. □

Solution 7: Down payment of a loan

Static and Dynamic Optimization

Notice, together with this solution comes (on the course home page) a distribution (dist1.zip) of m-files. On a unix system the distribution can be unpacked by the command: unzip -a dist1.zip.

1 Optimization

Just follow the instructions in the exercise.

2 Solving a set of equations

Just follow the instructions in the exercise.

3 Dynamic Optimization

Question: 1 We identify quite easily that:

$$N = 10, \quad x_0 = 50000, \quad \alpha = 0.05, \qquad a = 1 + \alpha = 1.05, \quad b = -1$$

and

$$f = ax_i + bu_i \qquad \phi = \frac{1}{2}px_N^2 \qquad L = \frac{1}{2}qx_i^2 + \frac{1}{2}ru_i^2$$

□

Question: 2

$$H_i = \frac{1}{2}qx_i^2 + \frac{1}{2}ru_i^2 + \lambda_{i+1}(ax_i + bu_i)$$

□

Question: 3

$$\frac{\partial}{\partial x}f = a \qquad \frac{\partial}{\partial x}L = qx_i$$

$$\frac{\partial}{\partial u}f = b \qquad \frac{\partial}{\partial u}L = ru_i$$

□

Question: 4 Solution given in the text. □

Question: 5 The stationarity condition (last equation) is simply:

$$u_i = -\frac{b}{r}\lambda_{i+1}$$

□

Question: 6 If we reverse the costate equation, then

$$\lambda_{i+1} = \frac{1}{a}\Big[\lambda_i - qx_i\Big]$$

□

Question: 7 Solution given in the text. □

Question: 8 The following code (fejlf.m) solves the recursions in (1).

```
function err=fejlf(la0,a,b,x0,p,r,q,N)
la=la0; x=x0;
for i=0:N-1,
 la=(la-q*x)/a;
 u=-b*la/r;
 x=a*x+b*u;
end
err=la-p*x;
```

The output is the error in the terminal boundary condition. □

Question: 9 The script below (runex3) uses fsolve for finding the correct initial costate ($\lambda(0)$ alias la0) such that the terminal boundary condition is fulfilled.

```
% Constants etc.
alf=0.05;
a=1+alf; b=-1;
x0=50000;
N=10;
q=alf^2; r=q; p=q;

%r=10*q;
%r=q/10;
%p=0;
%p=100*q;

% The search for la0
la0=10;
opt=optimset('fsolve');
opt=optimset(opt,'Display','off');
la0=fsolve('fejlf',la0,opt,a,b,x0,p,r,q,N)
```

```
% The simulation with the correct la0
ut=[];
la=la0; x=x0;
lat=la; xt=x;
for i=0:N-1,
 la=(la-q*x)/a;
 u=-b*la/r;
 x=a*x+b*u;
 xt=[xt;x]; lat=[lat;la]; ut=[ut;u];
end

subplot(211);
bar(ut); grid; title('Input sequence');
axis([0 15 0 50000]);
subplot(212);
bar(xt); grid; title('Balance');
axis([0 15 0 50000]);
```

□

Question: 10 Change the values in the script above (runex3.m) and run the script. □