$$\frac{\dot{c}}{c} = \frac{\lambda(t) - f}{\theta}$$

$$\dot{c}(t) = \frac{\lambda(t) - f}{\theta} c(t)$$

$$\dot{c}(t) - \frac{\lambda(t) - f}{\theta} c(t) = 0$$

$$\Rightarrow c(t) = C_0 \, e^{\frac{1}{\theta} \int_0^t (\lambda(s) - f) \, ds}$$

Using terminal condition,

$$2 = c(10) = C_0 \, e^{\frac{1}{\theta} \int_0^{10} (\lambda(s) - f) \, dx}$$

$$\lambda(s) = 0.05 + 0.01 t$$

$$\int_0^{10} (\lambda(s) - f) \, ds = \int_0^{10} (0.02 + 0.01 s) \, ds$$

$$= 0.02 (10) + 0.01 \frac{(100)}{2}$$

$$= 0.7$$

$$\Rightarrow \, \lambda = L_o \times e^{0.2/2}$$

$$L_o = \lambda \times e^{-\frac{0.7}{2}} = 1.4094$$

$$c(t) = 1.4094 \, e^{\frac{1}{2} \int_0^t (0.02 + 0.01s) \, ds}$$

$$= 1.4094 \, e^{\frac{1}{2} \left[0.02t + \frac{0.01}{2} t^2 \right]}$$

ProblemSetSolutions/PS1.m

```matlab
 1  % This is code for PS1 coding part
 2  % Author: Muhammad Bashir
 3  % Date: 24 October 2024
 4
 5
 6  % Parameters
 7  pars.C_T = 2;
 8  pars.r0 = 0.05;
 9  pars.alpha = 0.01;
10  pars.theta = 2
11  pars.rho = 0.03;
12  pars.T = 10;
13
14  % iterate to find vector r =r0+alha*t
15  r = zeros(pars.T,1);
16  r(1) = pars.r0;
17  for t = 2:pars.T
18      r(t) = pars.r0 + pars.alpha*t;
19  end
20
21  % Q.1 Solve Consumption equation analytically using integrating factor method
22  % C(t) = 1.4094*exp(0.5*(0.02t+0.01*t^2))
23
24  % Plot this solution function
25
26  t = 1:pars.T;
27  C = 1.4094*exp(0.5*(0.02*t+.5*0.01*t.^2));
28  figure
29  plot(t,C)
30  hold on
31  yline(2, '--r', 'LineWidth', 1.5, 'Label', 'Target Consumption at Time 10'); % Add
    horizontal line at y=2 with label
32  xlabel('Time (t)'); % Add x label
33  ylabel('Consumption (C)'); % Add y label
34  hold off
35
36  % Save picture
37  saveas(gcf, '/Users/muhammadbashir/GitHub/DynamicProgramming20↵
    24/ProblemSetSolutions/ConsumptionEvolutionAnalytical.png')
38
39  % Q.2 Solving numerically using Finite Difference Method
40  % create 1000 grid points between 0 and 10
41  pars.n = 100;
42  t = linspace(0, pars.T, pars.n);
43  pars.step_size = pars.T/pars.n;
44  % initialize consumption vector
45  C = zeros(pars.n,1);
46  % find (r(t)-rho)/theta for all t
47  discounter = (pars.r0 + pars.alpha*t - pars.rho)/pars.theta;
48  % initialize consumpsion at time T
49  C(pars.n) = 2;
50  % iterate backward to find consumption at all time periods
```

```matlab
51  for i = pars.n-1:-1:1
52      C(i) = C(i+1) - pars.step_size*(discounter(i)*C(i+1));
53  end
54
55  % Plot the solution along with the analytical solution
56  figure
57  scatter(t, C, 'o', 'LineWidth', .01) % Numerical solution as scatter plot with
    circles
58  hold on
59  plot(t, 1.4094*exp(0.5*(0.02*t+.5*0.01*t.^2)), '-r', 'LineWidth', 1) % Analytical
    solution as thick red line
60  xlabel('Time (t)'); % Add x label
61  ylabel('Consumption (C)'); % Add y label
62  yline(2, '--r', 'LineWidth', 1.5, 'Label', 'Target Consumption at Time 10'); % Add
    horizontal line at y=2 with label
63  legend('Numerical Solution', 'Analytical Solution','Location', 'northwest')
64  hold off
65  % save
66  saveas(gcf, '/Users/muhammadbashir/GitHub/DynamicProgramming20↩
    24/ProblemSetSolutions/ConsumptionEvolutionNumerical_100Steps.png')
67  % Store this C as C100
68  C100 = C;
69  t100 = t;
70  % Q.3 Repeat with 10 grid points
71  pars.n = 10;
72  t = linspace(0, pars.T, pars.n);
73  pars.step_size = pars.T/pars.n;
74  % initialize consumption vector
75  C = zeros(pars.n,1);
76  % find (r(t)-rho)/theta for all t
77  discounter = (pars.r0 + pars.alpha*t - pars.rho)/pars.theta;
78  % initialize consumption at time T
79  C(pars.n) = 2;
80  % iterate backward to find consumption at all time periods
81  for i = pars.n-1:-1:1
82      C(i) = C(i+1) - pars.step_size*(discounter(i)*C(i+1));
83  end
84
85  % Plot the solution along with the analytical solution
86  figure
87  scatter(t, C, 'o', 'LineWidth', .01) % Numerical solution as scatter plot with
    circles
88  hold on
89  scatter(t100, C100, 'x', 'LineWidth', .01) % Numerical solution as scatter plot with
    circles
90  plot(t, 1.4094*exp(0.5*(0.02*t+.5*0.01*t.^2)), '-r', 'LineWidth', 1) % Analytical
    solution as thick red line
91  xlabel('Time (t)'); % Add x label
92  ylabel('Consumption (C)'); % Add y label
93  yline(2, '--r', 'LineWidth', 1.5, 'Label', 'Target Consumption at Time 10'); % Add
    horizontal line at y=2 with label
94  legend('Numerical Solution (10 points)', 'Numerical Solution (100 points)',
    'Analytical Solution', 'Location', 'northwest')
95  hold off
96  % save
```

```matlab
97   saveas(gcf, '/Users/muhammadbashir/GitHub/DynamicProgramming20↵
     24/ProblemSetSolutions/ConsumptionEvolutionNumerical_10And100_Steps.png')
98   % Q.4 Discussion
99   % The numerical solution with 10 grid points is not as accurate as the one with 100
     grid points. As we see once we increase no of grid points, we reach closer to the
     analytical solution. This is because the finite difference method is an approximation
     method and the accuracy of the solution depends on the number of grid points used.
     The more the grid points, the more accurate the solution will be.
```