

1. We can rewrite the given expression in standard form,

$$\frac{\dot{C}(t)}{C(t)} = \frac{r_t - \rho}{\sigma} \Leftrightarrow \dot{C}(t) = \left[\frac{r_t - \rho}{\sigma} \right] C(t) \Leftrightarrow \dot{C}(t) - \left(\frac{r_t - \rho}{\sigma} \right) C(t) = 0. (*)$$

$$\text{Let } \mu(t) = e^{-\int \frac{r_t - \rho}{\sigma} dt} = e^{-\frac{1}{\sigma} \int (r_0 + \alpha t - \rho) dt} = e^{-\frac{1}{\sigma} [(r_0 - \rho)t + \frac{\alpha}{2} t^2]}$$

Multiplying (*) by $\mu(t)$ gives,

$$\mu(t) \dot{C}(t) - \mu(t) C(t) = 0 \mu(t)$$

$\Rightarrow \int \frac{d}{dt} [\mu(t) C(t)] dt = 0 dt$. Letting K denote the constant of integration,

$$\Rightarrow \mu(t) C(t) = K$$

$\Rightarrow e^{-\frac{1}{\sigma} [(r_0 - \rho)t + \frac{\alpha}{2} t^2]} C(t) = K$. Using $C(T) = C_T$ at $T=T$,

$\Rightarrow K = C_T e^{-\frac{1}{\sigma} [(r_0 - \rho)T + \frac{\alpha}{2} T^2]}$. The final solution is thus,

$$C(t) = K e^{\frac{1}{\sigma} [(r_0 - \rho)t + \frac{\alpha}{2} t^2]} = C_T e^{\frac{1}{\sigma} [(r_0 - \rho)t + \frac{\alpha}{2} t^2] - \frac{1}{\sigma} [(r_0 - \rho)T + \frac{\alpha}{2} T^2]}$$

$$\Rightarrow C(t) = C_T e^{\frac{1}{\sigma} [(r_0 - \rho)(t - T) + \frac{\alpha}{2} (t^2 - T^2)]}$$

$$\Rightarrow C(t) = \underline{C_T e^{\frac{1}{\sigma} [(r_0 - \rho)(t - T) + \frac{\alpha}{2} (t^2 - T^2)]}}$$

2. Given $\frac{\dot{C}(t)}{C(t)} = \frac{r_t - \rho}{\theta}$, we use the backwards difference method where $\dot{C}(t) = \frac{C(t) - C(t-\Delta t)}{\Delta t}$, which gives,

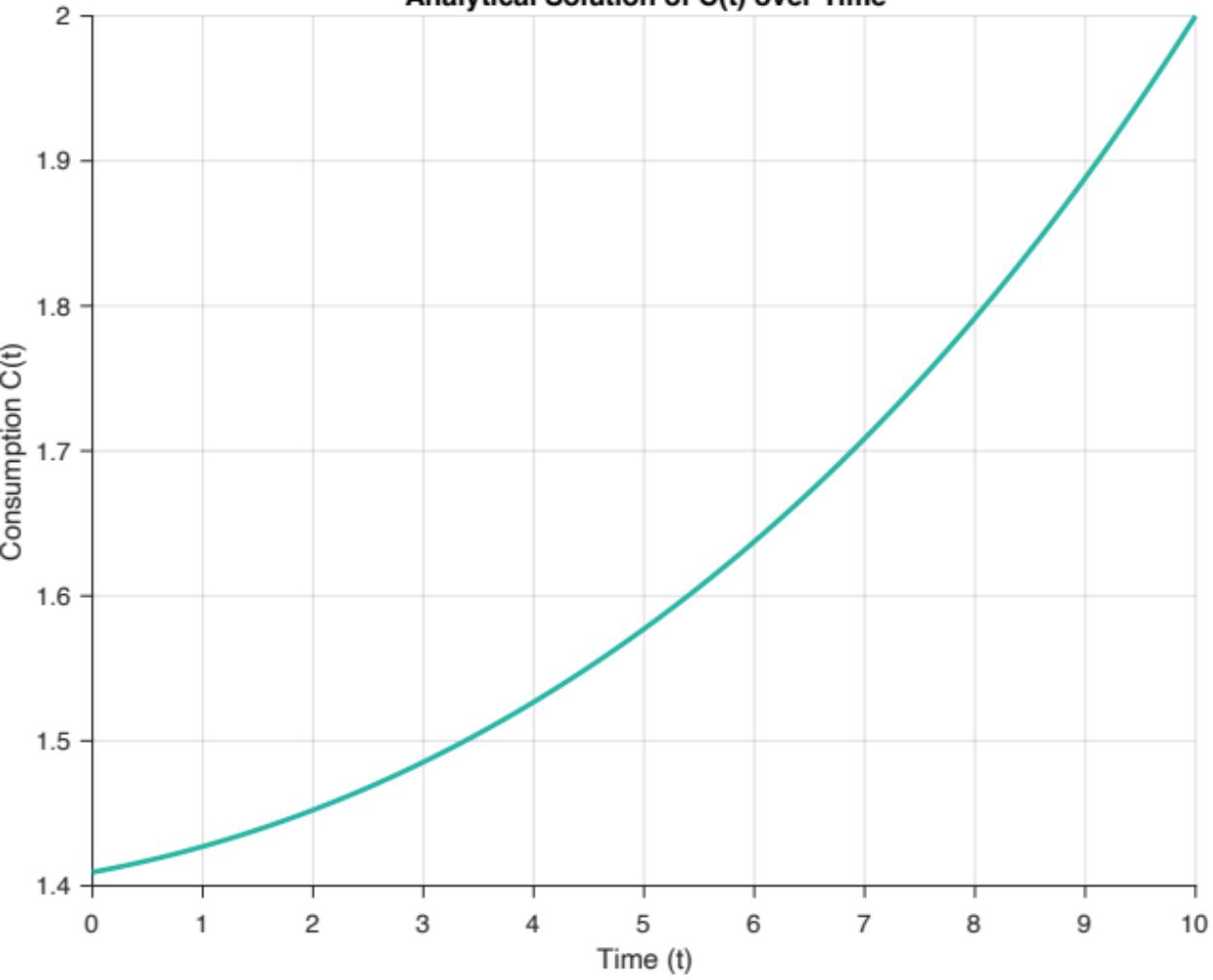
$$\frac{C(t) - C(t-\Delta t)}{\Delta t} \cdot \frac{1}{C(t)} = \frac{r_t - \rho}{\theta}$$

$$\Leftrightarrow C(t) - C(t-\Delta t) = \frac{\Delta t(r_t - \rho)}{\theta} C(t)$$

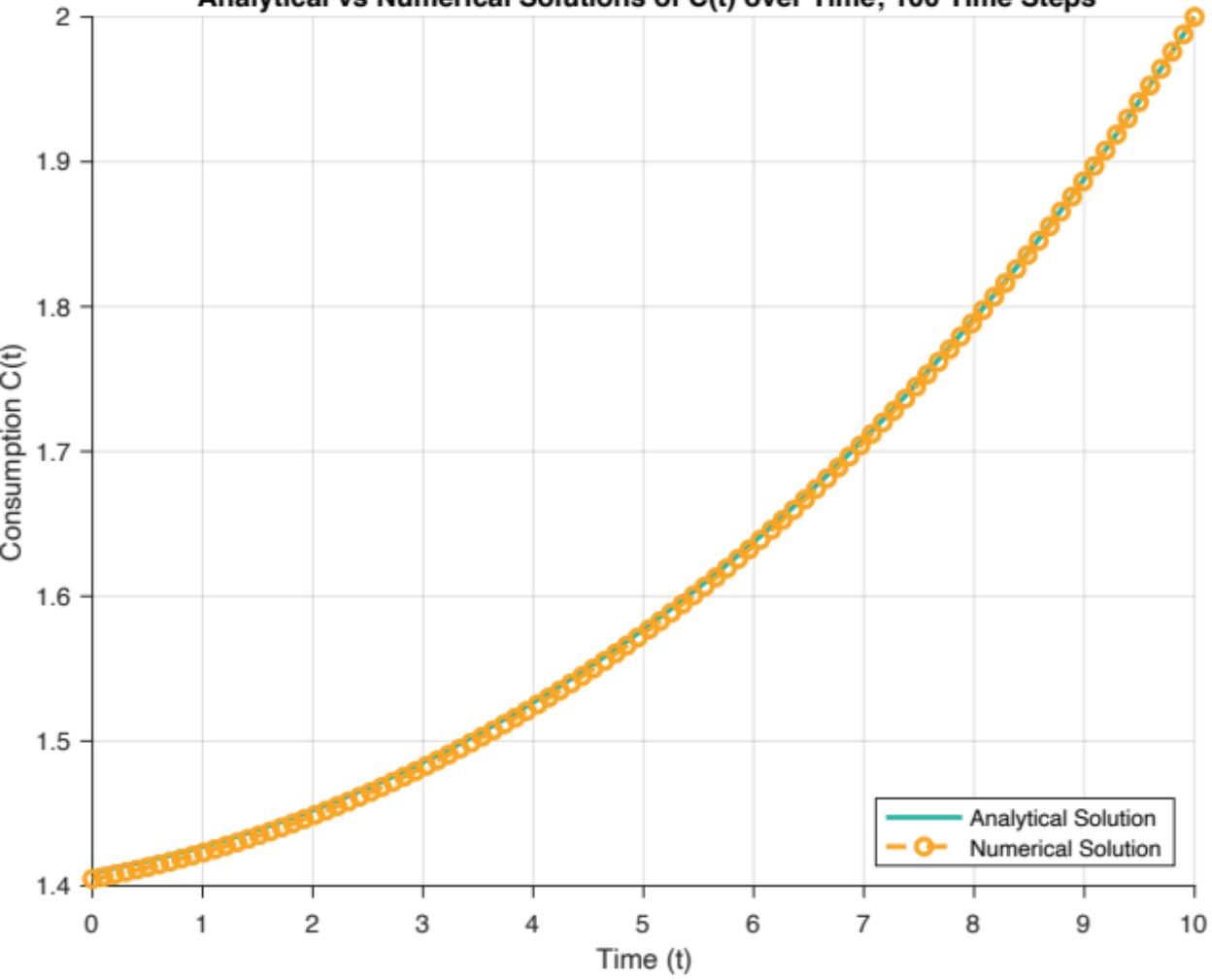
$$\Leftrightarrow C(t-\Delta t) = \underline{\left(\theta - \frac{\Delta t(r_t - \rho)}{\theta} \right) C(t)}$$

4. The numerical solution curve is much closer to the analytic solution curve in the 100 time-step case. For the 10 time-step case, they are noticeably different.

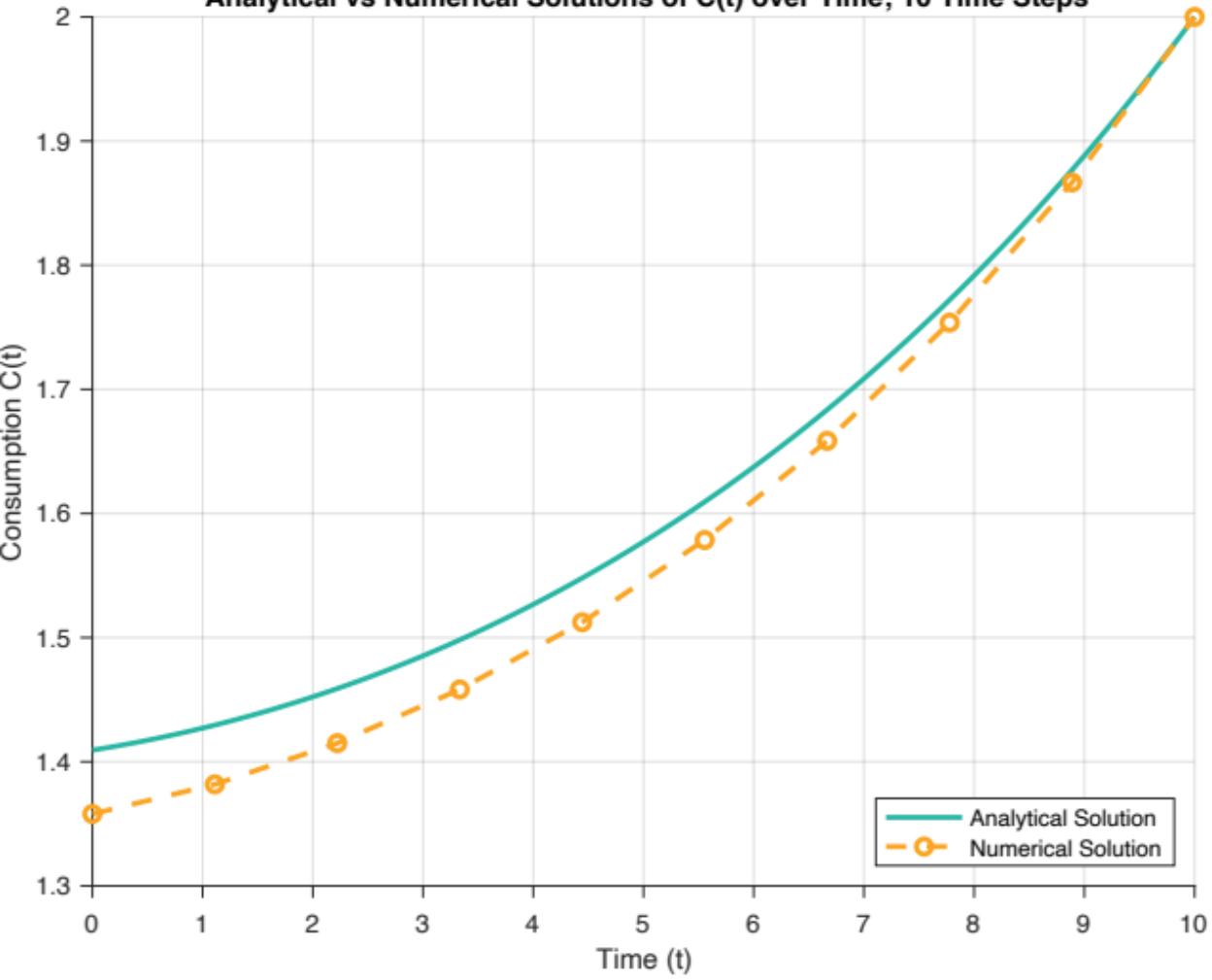
Analytical Solution of $C(t)$ over Time



Analytical vs Numerical Solutions of $C(t)$ over Time; 100 Time Steps



Analytical vs Numerical Solutions of $C(t)$ over Time; 10 Time Steps



```

~Documents/GitHub/DynamicProgramming2024/problem_sets/problem_set_1/problem_set_1_code.m
1 %{
2 -----
3 Purpose: Coding part of problem set 1
4 Created: Nico Rotundo 2024-10-28
5 -----
6 %}
7 %
8 %{
9 -----
10 Define parameters
11 -----
12 %}
13 %
14 % Terminal consumption at time T
15 C_T = 2.0;
16 %
17 % Initial interest rate
18 r_0 = 0.05;
19 %
20 % Growth rate of interest rate
21 alpha = 0.01;
22 %
23 %{
24 % Interest rate function as a handle function of time t
25 function r = interest_rate_function(t, r_0, alpha)
26     r = r_0 + alpha * t;
27 end
28 %}
29 %
30 % Inverse of the intertemporal elasticity of substitution
31 theta = 2.0;
32 %
33 % Time discount factor
34 rho = 0.03;
35 %
36 % Terminal time
37 T = 10;
38 %
39 % Display the result of interest rate function at specific time to check
40 %fprintf('The interest rate at t = %d is: %.3f\n', 20, interest_rate_function(20, r_0, alpha));
41 %
42 % Checking how assert statements work in this matlab
43 %assert(interest_rate_function(20, r_0, alpha) == .25);
44 %
45 %
46 %{
47 -----
48 1. Solve the equation analytically using the integrating factor method. Then, plot the analytical solution as a function
49 of time t and consumption C(t)
50 -
51 solved equation: C(t) = C_T*exp([(1/theta)*((r_0-rho)*(t-T)+(alpha/2)*(t^2-T^2))]
52 %
53 %}
54 %
55 % Time vector from 0 to T with 100 points
56 t = linspace(0, T, 100);
57 %
58 % Define value for consumption at time t
59 C_t_analytical = C_T * exp((1 / theta) * ((r_0 - rho) * (t - T) + (alpha / 2) * (t.^2 - T.^2)));
60 %
61 % Generate figure for analytical solution
62 figure;
63 plot(t, C_t_analytical, 'LineWidth', 2, 'Color', [41/255, 182/255, 164/255]);
64 xlabel('Time (t)');
65 set(gca, 'TickDir', 'out');
66 box off;
67 ylabel('Consumption C(t)');
68 title('Analytical Solution of C(t) over Time');
69 grid on;
70 %
71 % Export
72 exportgraphics(gcf, '/Users/nicorotundo/Documents/GitHub/DynamicProgramming2024/problem_sets/problem_set_1/output/question_1.pdf', 'ContentType', 'vector');
73 %
74 %{
75 -----
76 2. Solve the equation numerically using the finite difference method with 100 time steps. Plot the numerical and analytical
77 solutions together in a single figure for comparison.
78 -
79 equation: C(t-dt) = (theta*dt*(r_t-rho))/theta * C(t)
80 %
81 %}
82 %
83 % Time step (\triangle t)
84 dt = 10/99;
85 %
86 % Number of time steps
87 N = 100;
88 %
89 % Vector to store consumption values
90 C_t_numeric = zeros(N, 1);
91 %
92 % Set terminal consumption
93 C_t_numeric(end) = C_T;
94 %
95 % Time vector from 0 to T with 100 points
96 t = linspace(0, T, N);
97 %
98 % Time stepping using the finite difference method
99 for n = N:-1:2
100 %
101 % Compute consumption at the next time step using backward difference
102 C_t_numeric(n-1) = (theta - dt*(r_0 + alpha*t(n)-rho))/theta * C_t_numeric(n);
103 end
104 %
105 % Generate figure for numeric solution
106 figure;
107 plot(t, C_t_numeric, 'LineWidth', 2, 'Color', [250/255, 165/255, 35/255]);
108 xlabel('Time (t)');
109 set(gca, 'TickDir', 'out');
110 box off;
111 ylabel('Consumption C(t)');
112 title('Numerical Solution of C(t) over Time');
113 grid on;
114 %
115 % Generate both analytical and numeric sultions on the same plot
116 figure;
117 %
118 % Plot analytical solution
119 plot(t, C_t_analytical, 'LineWidth', 2, 'Color', [41/255, 182/255, 164/255]);
120 hold on;
121 %
122 % Plot numerical solution
123 plot(t, C_t_numeric, 'bo--', 'LineWidth', 2, 'Color', [250/255, 165/255, 35/255]);
124 %
125 % Add labels and title
126 xlabel('Time (t)');
127 ylabel('Consumption C(t)');
128 title('Analytical vs Numerical Solutions of C(t) over Time; 100 Time Steps');
129 %
130 % Add a legend
131 legend({'Analytical Solution', 'Numerical Solution'}, 'Location', 'southeast');
132 %
133 % Adjust plot settings
134 set(gca, 'TickDir', 'out');
135 box off;
136 grid on;
137 %
138 hold off;
139 %
140 % Export
141 exportgraphics(gcf, '/Users/nicorotundo/Documents/GitHub/DynamicProgramming2024/problem_sets/problem_set_1/output/question_2.pdf', 'ContentType', 'vector');
142 %
143 %{
144 -----
145 3. Repeat the numerical solution using 10 time steps. Plot the numerical and analytical solutions together in a single
146 figure.
147 %
148 %}
149 %
150 %
151 % Time step (\triangle t)
152 dt = 10/9;
153 %
154 % Number of time steps
155 N = 10;
156 %
157 % Vector to store consumption values
158 C_t_numeric = zeros(N, 1);
159 %
160 % Set terminal consumption
161 C_t_numeric(end) = C_T;
162 %
163 % Time vector from 0 to T with 10 points
164 t_10 = linspace(0, T, N);
165 %
166 % Time stepping using the finite difference method
167 for n = N:-1:2
168 %
169 % Compute consumption at the next time step using backward difference
170 C_t_numeric(n-1) = (theta - dt*(r_0 + alpha*t_10(n)-rho))/theta * C_t_numeric(n);
171 end
172 %
173 % Generate both analytical and numeric sultions on the same plot
174 figure;
175 %
176 % Plot analytical solution
177 plot(t, C_t_analytical, 'LineWidth', 2, 'Color', [41/255, 182/255, 164/255]);
178 hold on;
179 %
180 % Plot numerical solution
181 plot(t_10, C_t_numeric, 'bo--', 'LineWidth', 2, 'Color', [250/255, 165/255, 35/255]);
182 %
183 % Add labels and title
184 xlabel('Time (t)');
185 ylabel('Consumption C(t)');
186 title('Analytical vs Numerical Solutions of C(t) over Time; 10 Time Steps');
187 %
188 % Add a legend
189 legend({'Analytical Solution', 'Numerical Solution'}, 'Location', 'southeast');
190 %
191 % Adjust plot settings
192 set(gca, 'TickDir', 'out');
193 box off;
194 grid on;
195 %
196 hold off;
197 %
198 % Export
199 exportgraphics(gcf, '/Users/nicorotundo/Documents/GitHub/DynamicProgramming2024/problem_sets/problem_set_1/output/question_3.pdf', 'ContentType', 'vector');
200 %
201

```