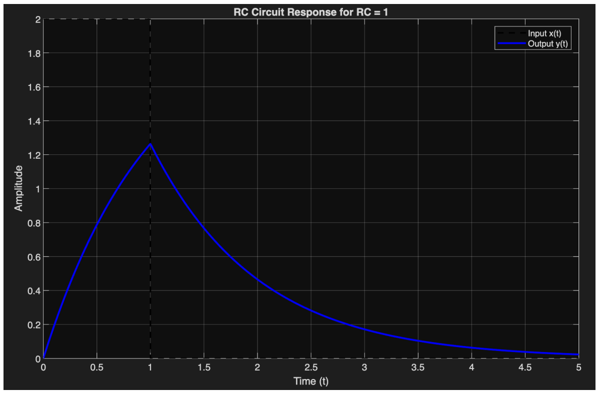
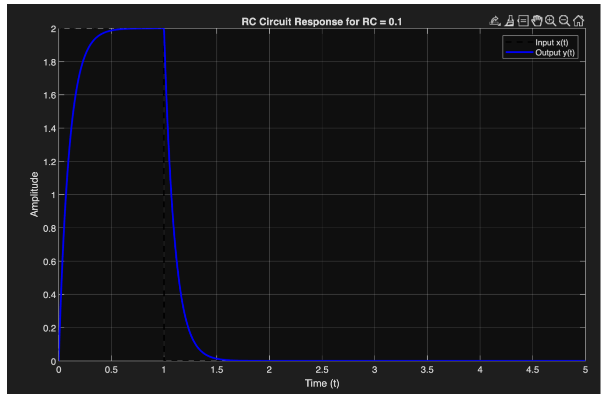
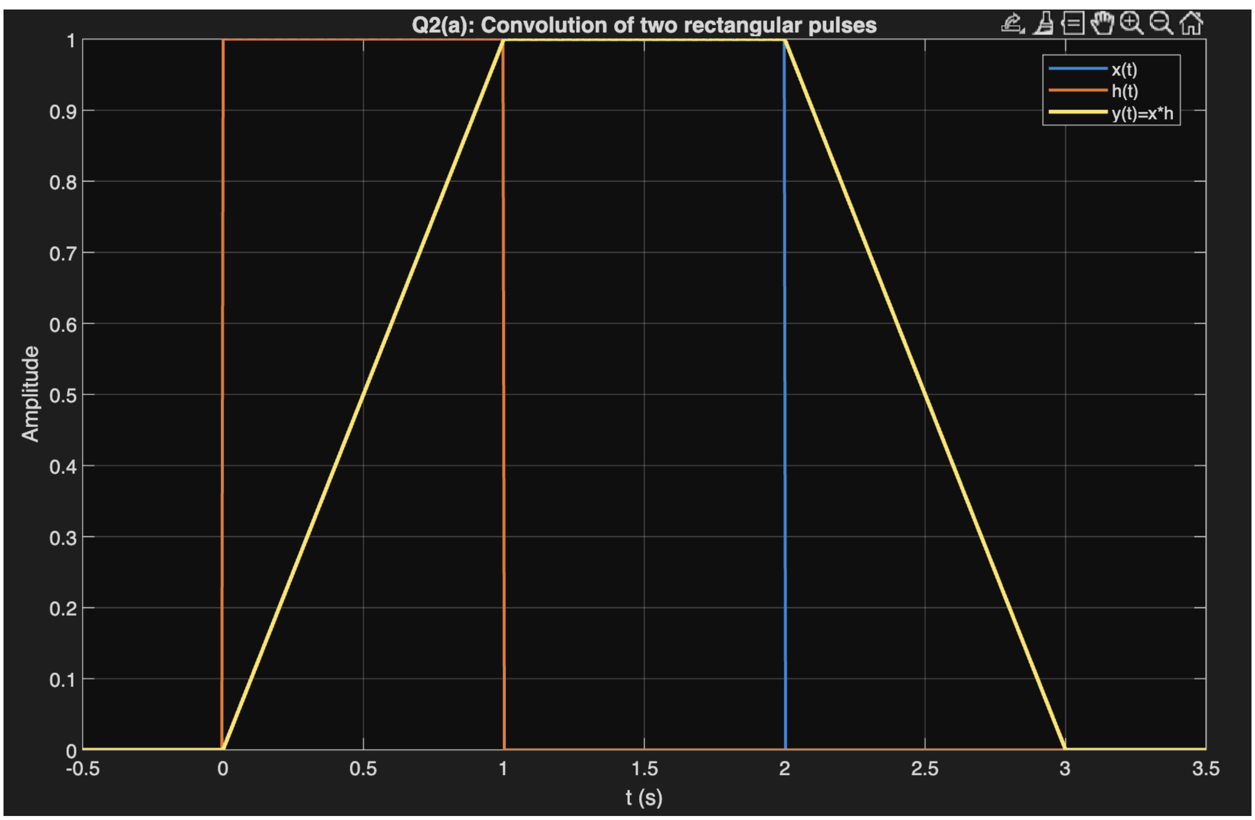
Problem 1. a & b





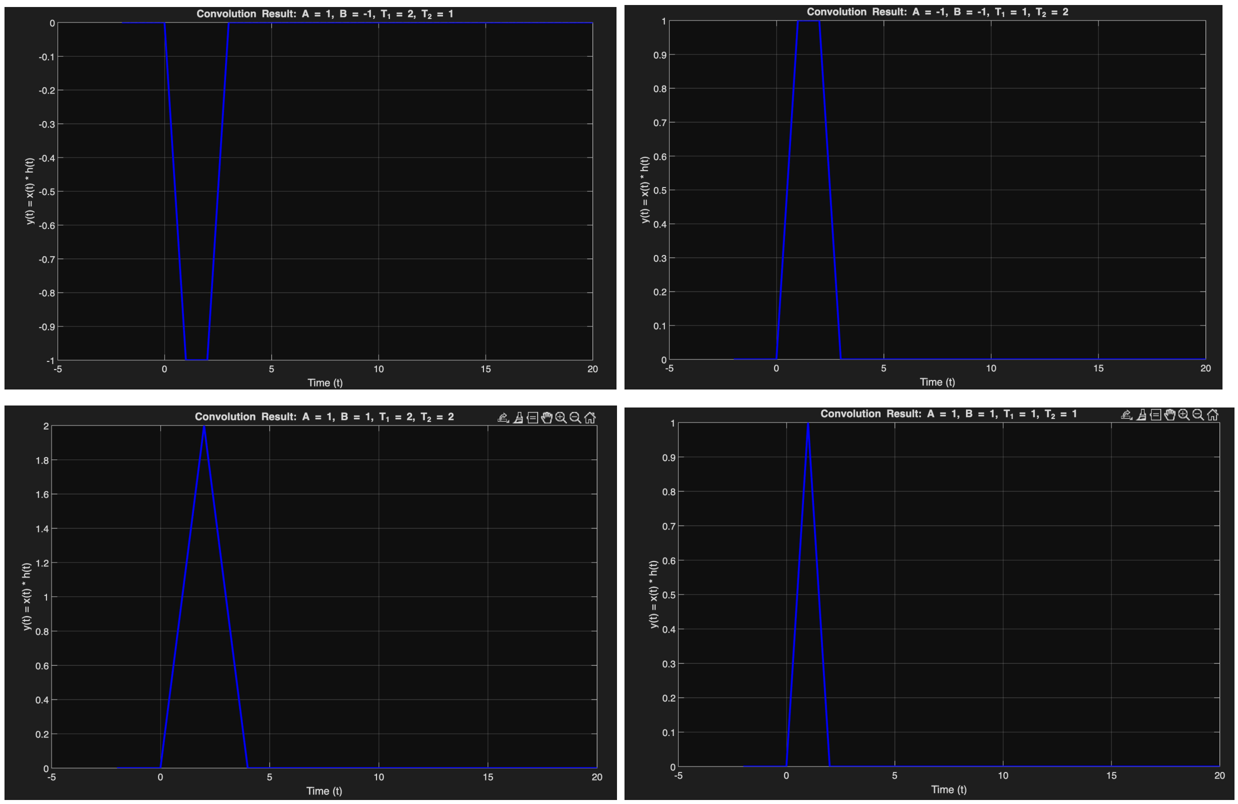
The MATLAB code computes y(t) analytically by dividing the time vector t = 0 : 0.005 : 5s into two regions, using logical indexing to apply the correct equation for each interval. For $0 \leq t < 1$, it calculates $y(t) = 2(1-e^{-t/RC})$ at t = 1. Continuity is enforced with $y(1) = 2(1-e^{-1/RC})$, which is then used for $t \geq 1$ where $y(t) = y(1)e^{-(t-1)/RC}$. The script stores these values in arrays for each RC value, being 0.1 and 1 seconds, producing plots that show the rectangular input pulse x(t) and the exponential rise and decay of y(t).

Problem 2. A



The convolution y(t) = x(t) * h(t) is derived by finding the overlap of the two rectangular pulses x(t) = A[u(t) - u(t-T_1)] and h(t) = B[u(t) - u(t-T_2)]. The overlap length is [min(T_1,t) - max(0, t-T_2)]. Multiplying A and B gives you y(t) = AB[min(T_1,t) - max(0, t-T_2)]. This simplifies to a trapezoidal or triangular waveform defined piecewise over breakpoints 0, min(T_1,T_2), max(T_1,T_2), T_1 + T_2.

Problem 2. B



This MATLAB code computes y(t) = x(t) * h(t) numerically by defining x(t) and h(t) as rectangular pulses over a time vector from -1 seconds to 10 seconds with a step size of 0.01 seconds. It uses MATLAB's conv function, scaled by the tiem step to approximate the convolution integral for each of the test cases defined by the different values of A, B, T_1, and T_2. The script then generates plots of y(t) for each case. This shows the rapezoidal or triangular shape predicted by the analytic solution in part 2a.

Appendix

Problem 1. a & b

```
% ELCT 222 --- Computer Assignment 2
% Problem 1: Analytical solution of RC circuit response to a rectangular pulse
input
% Define a time vector from 0 to 5 seconds with 100 evenly spaced points.
t = linspace(0, 5, 1000);
% Define the input x(t): a rectangular pulse of height 2 from
% t = 0 to t = 1
% Use logical indexing: (t >= 0 & t < 1) evaluates to 1 in that interval
% and 0 elsewhere
x = 2 * (t >= 0 & t < 1);
% Define the two RC values to analyze:
% Case 1: RC = 0.1 --> faster system
% Case 2: RC = 1   --> slower system
RC_values = [0.1, 1];
% Preallocate a matrix to store output responses for each RC case
% Each row of y_outputs will hold y(t) for one RC value
y_outputs = zeros(length(RC_values), length(t));
% Loop over each RC value to compute and plot the corresponding output y(t)
for k = 1:length(RC_values)
   RC = RC_values(k);  % Current RC value
   % Initialize output vector y(t) to all zeros for this case
   y = zeros(size(t));
   % --- FIRST INTERVAL: 0 ≤ t < 1 ---
   % During this time, the input x(t) = 2 (constant)
   % Solve: RC * dy/dt + y = 2
   % Homogeneous solution + particular solution gives:
   % y(t) = 2(1 - exp(-t / RC)) with initla condition y(0) = 0
   idx1 = t < 1;  % Indices where t < 1
   y(idx1) = 2 * (1 - exp(-t(idx1) / RC));
   % --- SECOND INTERVAL: t ≥ 1 ---
   % At t = 1, the input x(t) drops to 0.
   % Now solve: RC * dy/dt + y = 0
   % Use y(1) from the previous step as initial condition for continuity
   y1 = 2 * (1 - exp(-1 / RC));  % This is y(t=1)
   % Output continues as exponential decay:
   % y(t) = y(1) * exp(-(t - 1) / RC) for t ≥ 1
   idx2 = t >= 1;  % Indices where t ≥ 1
   y(idx2) = y1 * exp(-(t(idx2) - 1) / RC);
   % --- PLOTTING ---
   figure;
   plot(t, x, 'k--', 'LineWidth', 1.5); hold on; % Input x(t) in dashed black
   plot(t, y, 'b-', 'LineWidth', 2);  % Output y(t) in solid blue
   % Label and title
   xlabel('Time (t)');
   ylabel('Amplitude');
   title(['RC Circuit Response for RC = ', num2str(RC)]);
   % Legend and grid for clarity
   legend('Input x(t)', 'Output y(t)');
   grid on;
```

```
end
```

Problem 2. a

```
% ELCT 222 --- CA2-2a (minimal)
clear; clc; close all;
A = 1; B = 1; T1 = 2; T2 = 1;           % set the case you need
yfun = @(t) A*B*max(0, min(T1, t) - max(0, t - T2));  % analytic result
t = linspace(-0.5, T1+T2+0.5, 800);
x = A*(t>=0 & t<T1);
h = B*(t>=0 & t<T2);
y = yfun(t);
plot(t,x,'LineWidth',1.5); hold on;
plot(t,h,'LineWidth',1.5);
plot(t,y,'LineWidth',2);
grid on; xlabel('t (s)'); ylabel('Amplitude');
legend('x(t)','h(t)','y(t)=x*h','Location','best');
title('Q2(a): Convolution of two rectangular pulses');
```

Problem 2. b

```
% ELCT 222 --- Computer Assignment 2
% Problem 2(b): MATLAB convolution of two rectangular pulses
% Define time range
t = -1:0.01:10;
% Define the four test cases: [A, B, T1, T2]
cases = [
   1, 1, 1, 1;
   1, 1, 2, 2;
   -1, -1, 1, 2;
   1, -1, 2, 1;
];
% Loop over all 4 cases
for i = 1:4
   % Extract parameters
   A = cases(i, 1);
   B = cases(i, 2);
   T1 = cases(i, 3);
   T2 = cases(i, 4);
   % Define x(t) = A * [u(t) - u(t - T1)]
   x = A * ((t >= 0) & (t < T1));
   % Define h(t) = B * [u(t) - u(t - T2)]
   h = B * ((t >= 0) & (t < T2));
   % Perform convolution
   y = conv(x, h) * 0.01;  % Multiply by dt to approximate integral
   % Time vector for convolution result
   t_y = 2 * t(1) + 0.01 * (0:length(y)-1);
   % Plot result
   figure;
   plot(t_y, y, 'b', 'LineWidth', 2);
   xlabel('Time (t)');
```

```
    ylabel('y(t) = x(t) * h(t)');
    title(sprintf('Convolution Result: A = %d, B = %d, T_1 = %d, T_2 = %d', A,
B, T1, T2));
    grid on;
end
```