EXERCISE-1

1. **Continuous-Time Unit Impulse Signal (Approximated)**

clc;

clear;

t = -1:0.001:1;

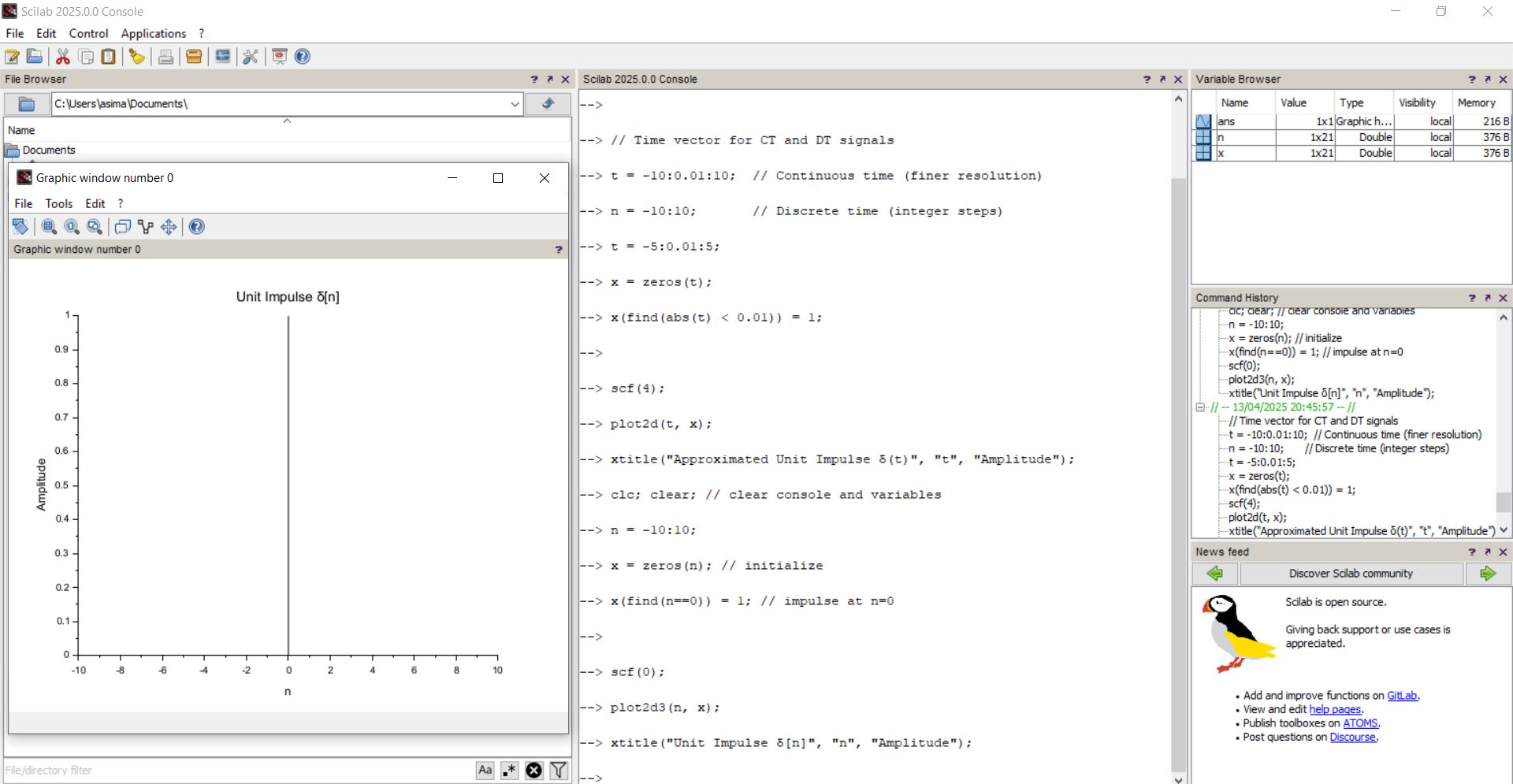
delta = zeros(1, length(t));

delta(find(abs(t) < 0.005)) = 1/0.01; // Area ≈ 1

scf(4);

plot(t, delta);

xtitle("Continuous-Time Unit Impulse (Approximated)", "t", "Amplitude");



1. Continuous-Time Unit Step Signal

clc;

clear;

t = -2:0.01:2;

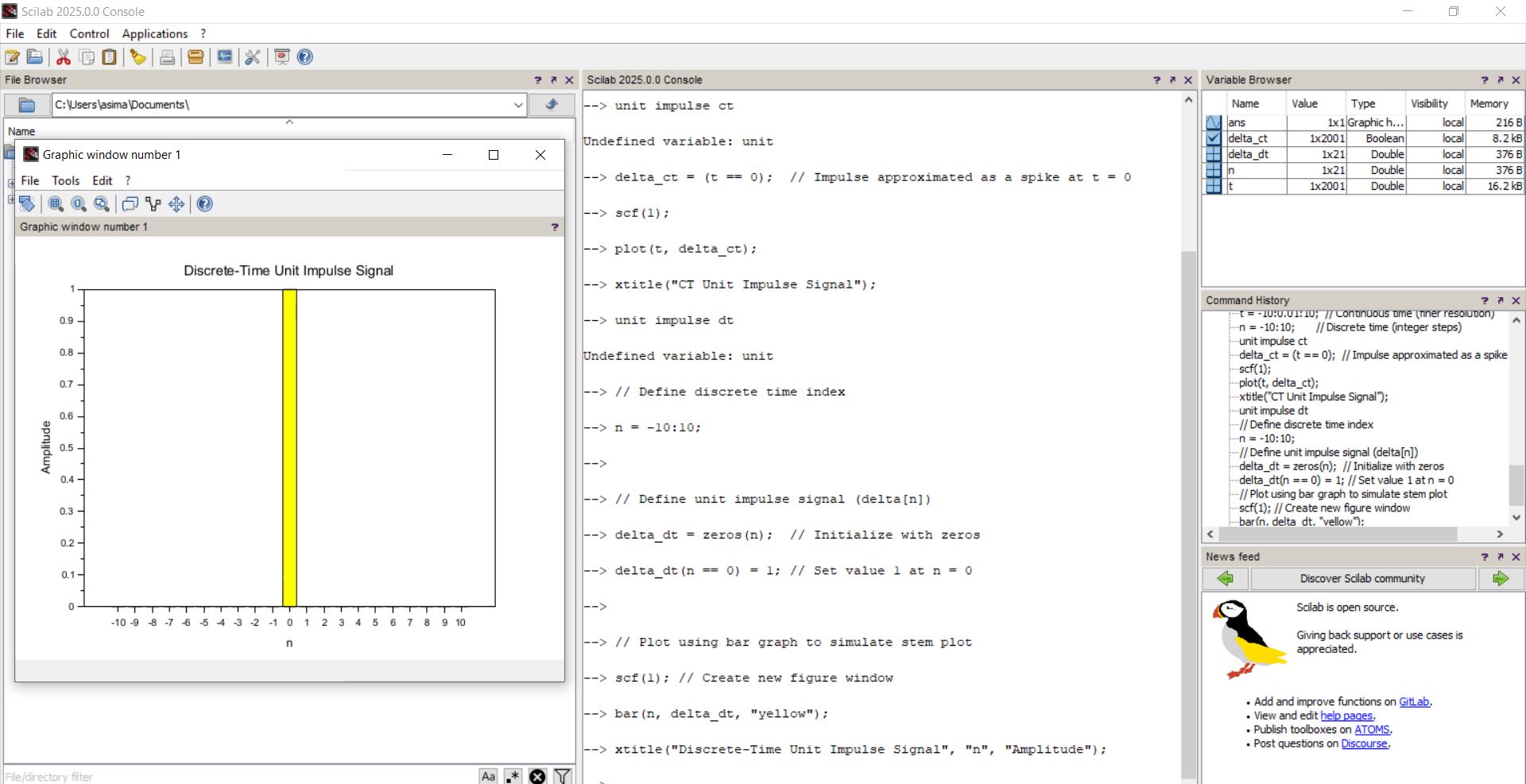
u\_ct = zeros(1, length(t));

u\_ct(find(t >= 0)) = 1;

scf(5);

plot(t, u\_ct);

xtitle("Continuous-Time Unit Step Signal", "t", "Amplitude");



1. Continuous-Time Sinusoidal Signal

clc;

clear;

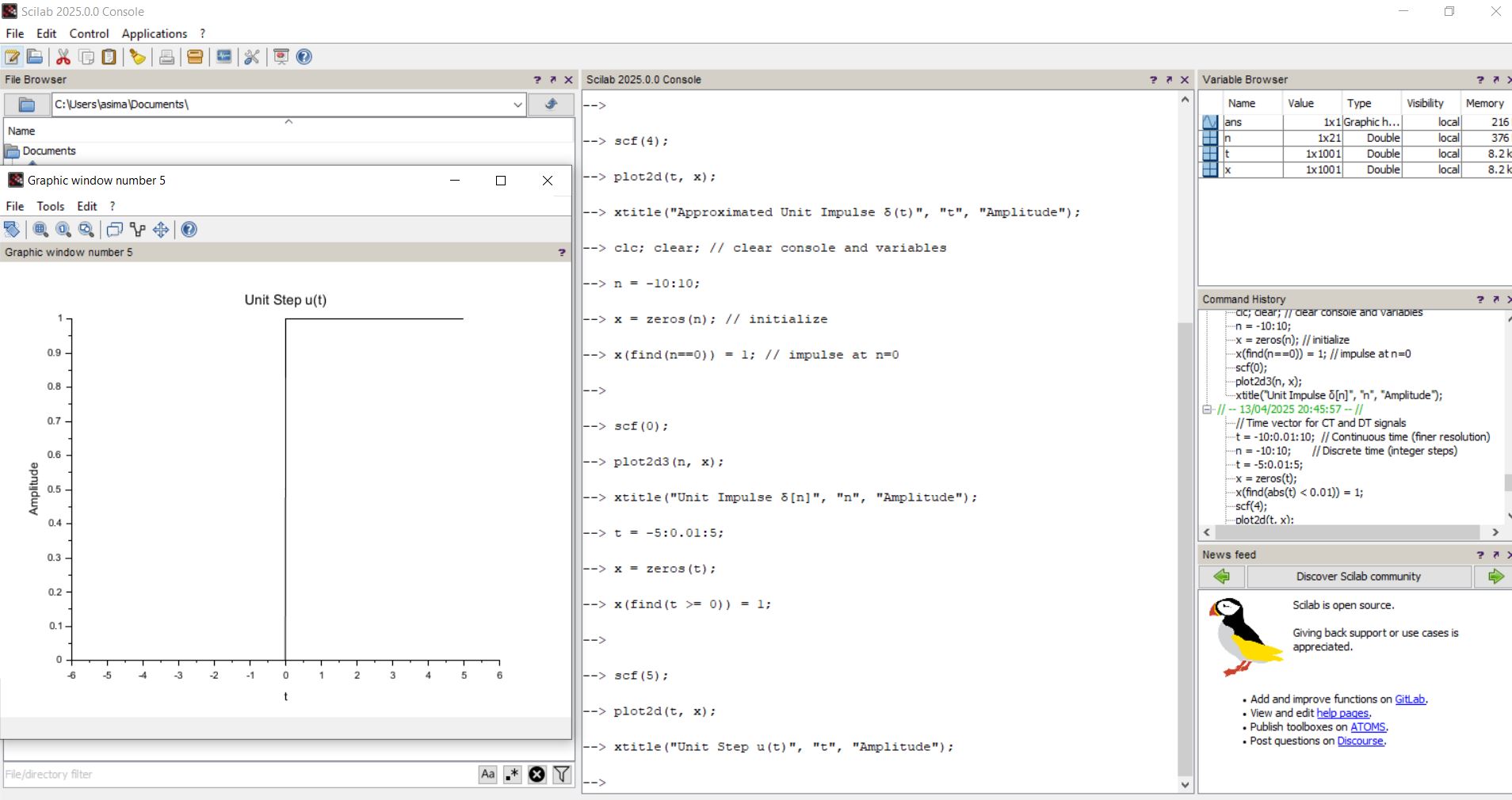
t = 0:0.01:5;

x\_ct\_sin = sin(2 \* %pi \* t); // 1 Hz sinusoid

scf(6);

plot(t, x\_ct\_sin);

xtitle("Continuous-Time Sinusoidal Signal", "t", "Amplitude");



1. Continuous-Time Exponential Signal

clc;

clear;

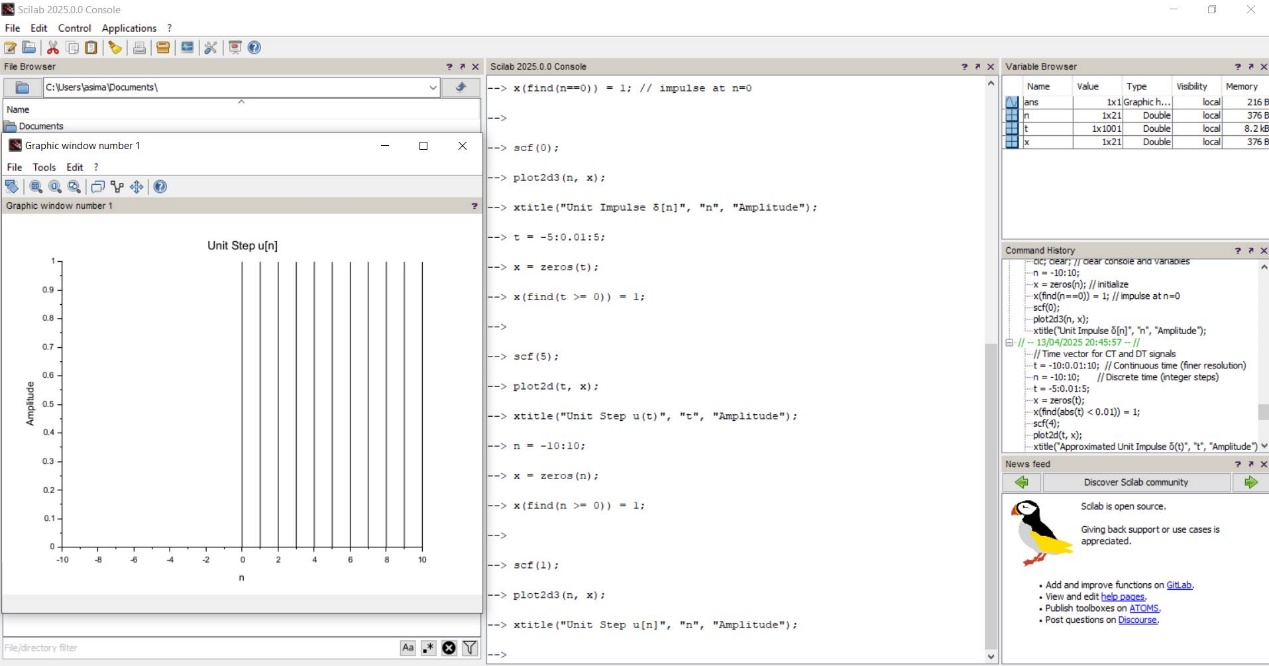
t = 0:0.01:5;

x\_ct\_exp = exp(-t); // Exponential decay

scf(7);

plot(t, x\_ct\_exp);

xtitle("Continuous-Time Exponential Signal", "t", "Amplitude");



1. Discrete-Time Unit Impulse Signal

clc;

clear;

close();

n = -10:10;

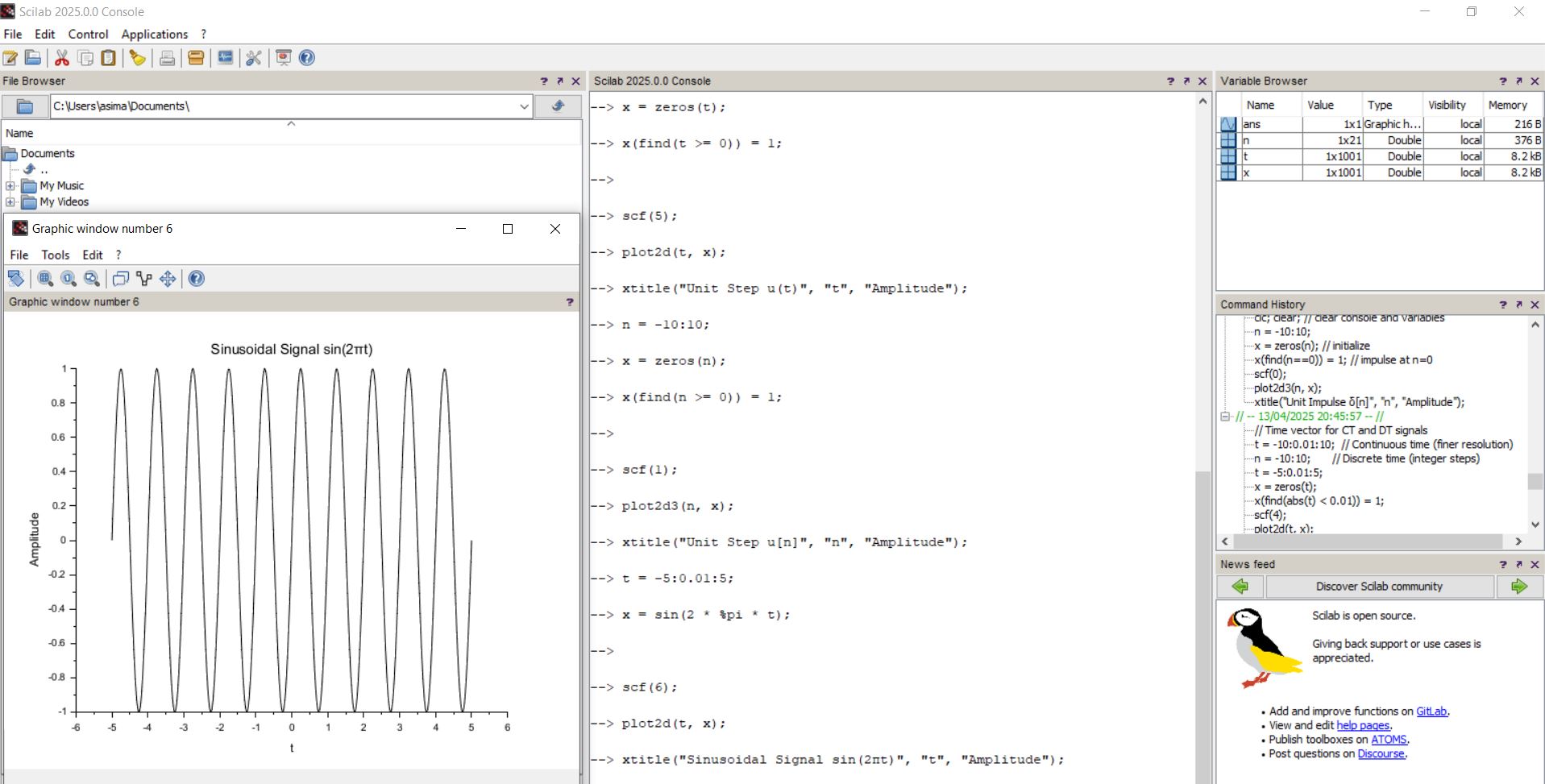
imp = zeros(1, length(n));

imp(find(n == 0)) = 1; // Set value at n = 0 to 1

scf(0);

plot2d3(n, imp);

xtitle("Discrete-Time Unit Impulse Signal", "n", "Amplitude");



1. Discrete-Time Unit Step Signal

clc;

clear;

close();

n = -10:10;

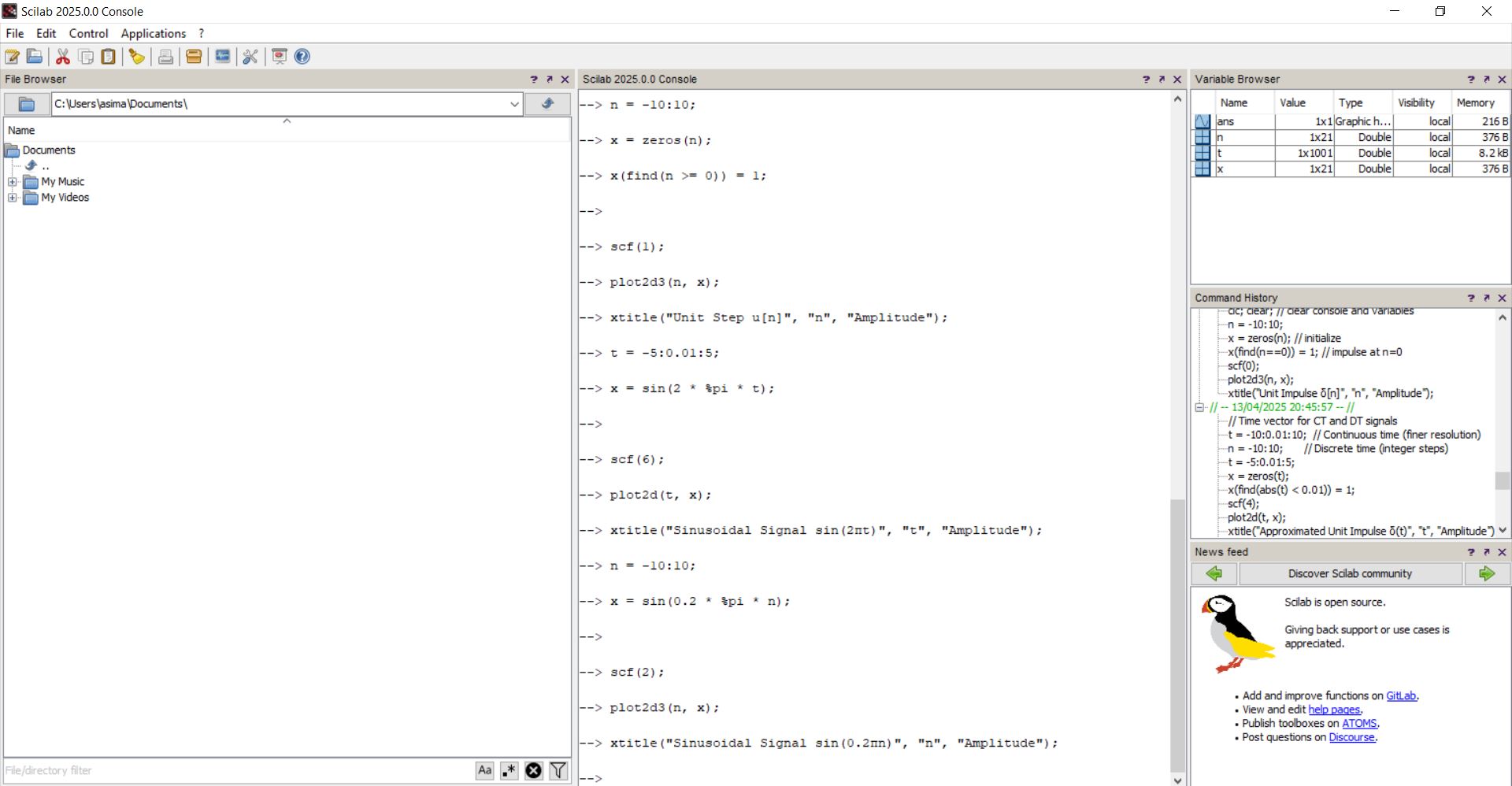
u = zeros(1, length(n));

u(find(n >= 0)) = 1; // Set values for n ≥ 0 to 1

scf(1);

plot2d3(n, u);

xtitle("Discrete-Time Unit Step Signal", "n", "Amplitude");



1. Discrete-Time Sinusoidal Signal

clc;

clear;

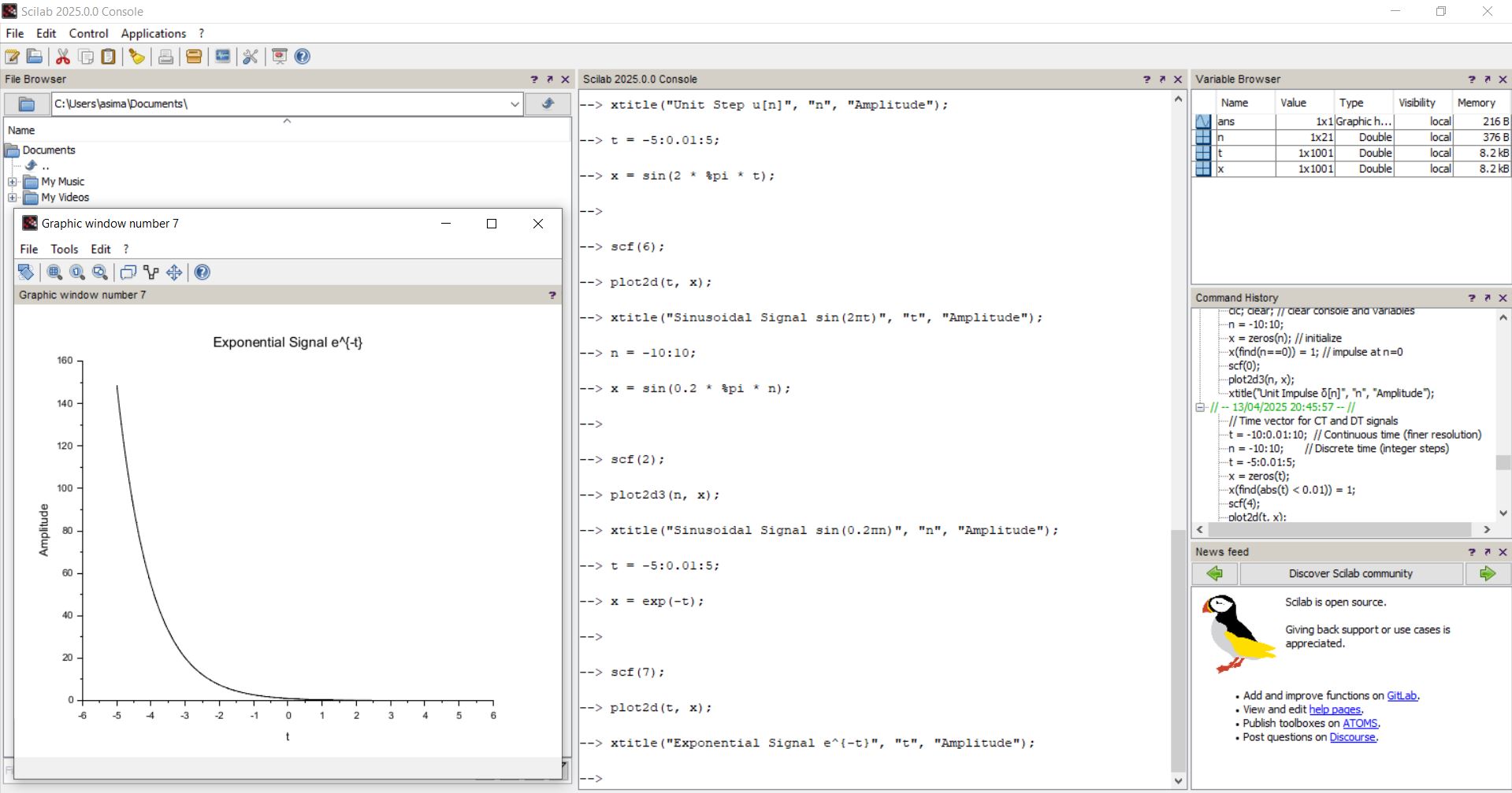
n = 0:40;

x\_sin = sin(%pi/4 \* n); // Discrete sinusoid with ω = π/4

scf(2);

plot2d3(n, x\_sin);

xtitle("Discrete-Time Sinusoidal Signal", "n", "Amplitude");



1. Discrete-Time Exponential Signal

clc;

clear;

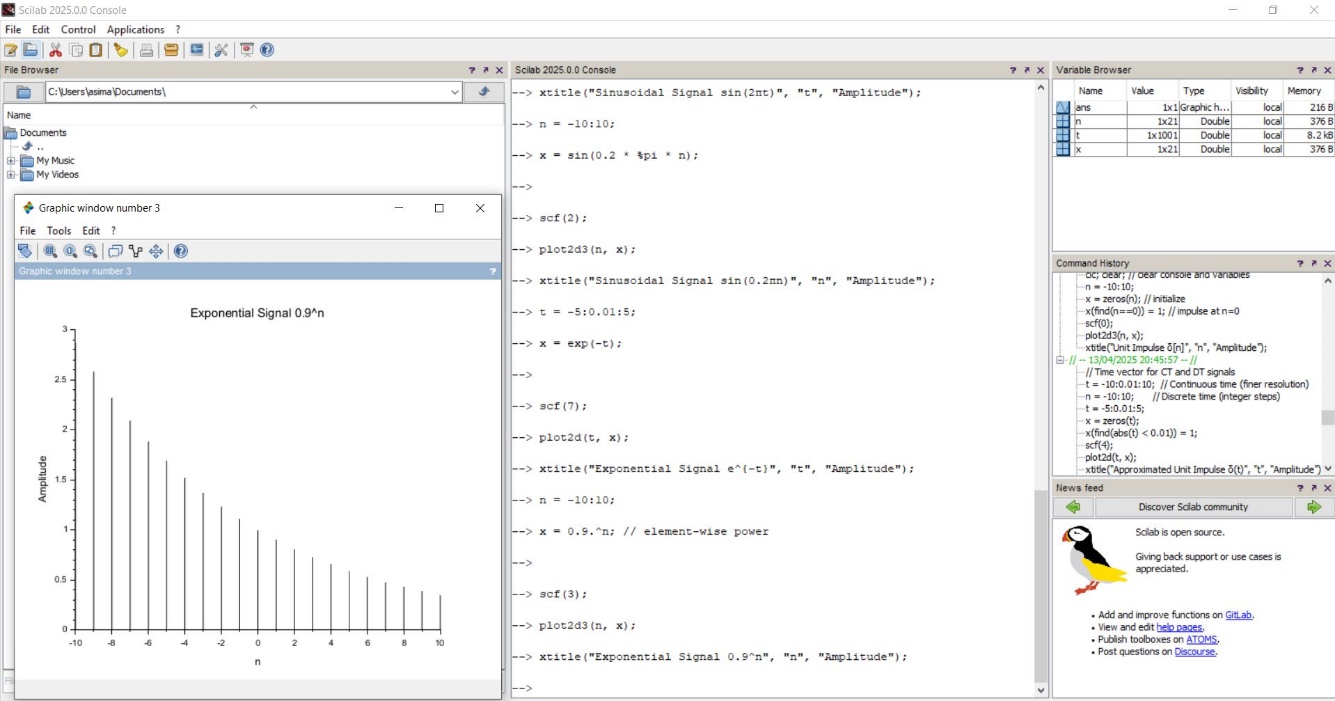
n = 0:20;

x\_exp = (0.8).^n; // Decaying exponential

scf(3);

plot2d3(n, x\_exp);

xtitle("Discrete-Time Exponential Signal", "n", "Amplitude");



EXERCISE-2

**Signals Used:**

* x1[n] = sin(π/4 \* n)
* x2[n] = cos(π/4 \* n)

1. CT SIGNAL

clc;

clear;

close();

t = 0:0.01:5; // Time range

x1 = sin(2 \* %pi \* t); // x1(t) = sin(2πt)

x2 = exp(-t); // x2(t) = exp(-t)

// Addition of signals

x\_add = x1 + x2;

// Multiplication of signals (pointwise)

x\_mult = x1 .\* x2;

// Plotting

Scf(2);

subplot(4, 1, 1);

plot(t, x1);

xtitle("x1(t) = sin(2πt)", "t", "Amplitude");

subplot(4, 1, 2);

plot(t, x2);

xtitle("x2(t) = exp(-t)", "t", "Amplitude");

subplot(4, 1, 3);

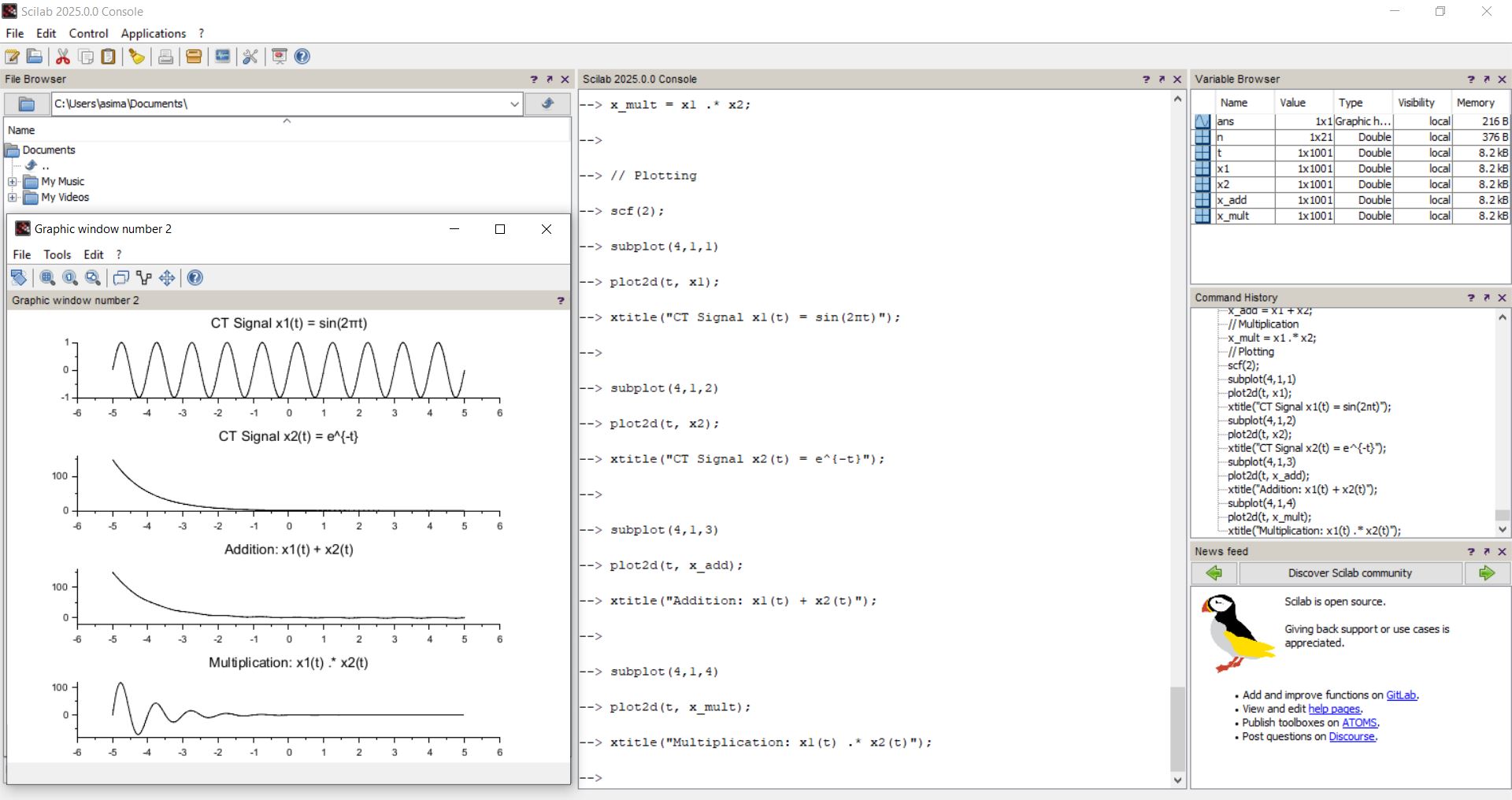
plot(t, x\_add);

xtitle("x1(t) + x2(t) (Addition)", "t", "Amplitude");

subplot(4, 1, 4);

plot(t, x\_mult);

xtitle("x1(t) .\* x2(t) (Multiplication)", "t", "Amplitude");



1. DT SIGNAL

scf(2);

subplot(4,1,1);

plot2d3(n, x1\_dt);

xtitle("DT: x1[n] = cos(0.4πn)", "n", "Amplitude");

subplot(4,1,2);

plot2d3(n, x2\_dt);

xtitle("DT: x2[n] = 0.8^n", "n", "Amplitude");

subplot(4,1,3);

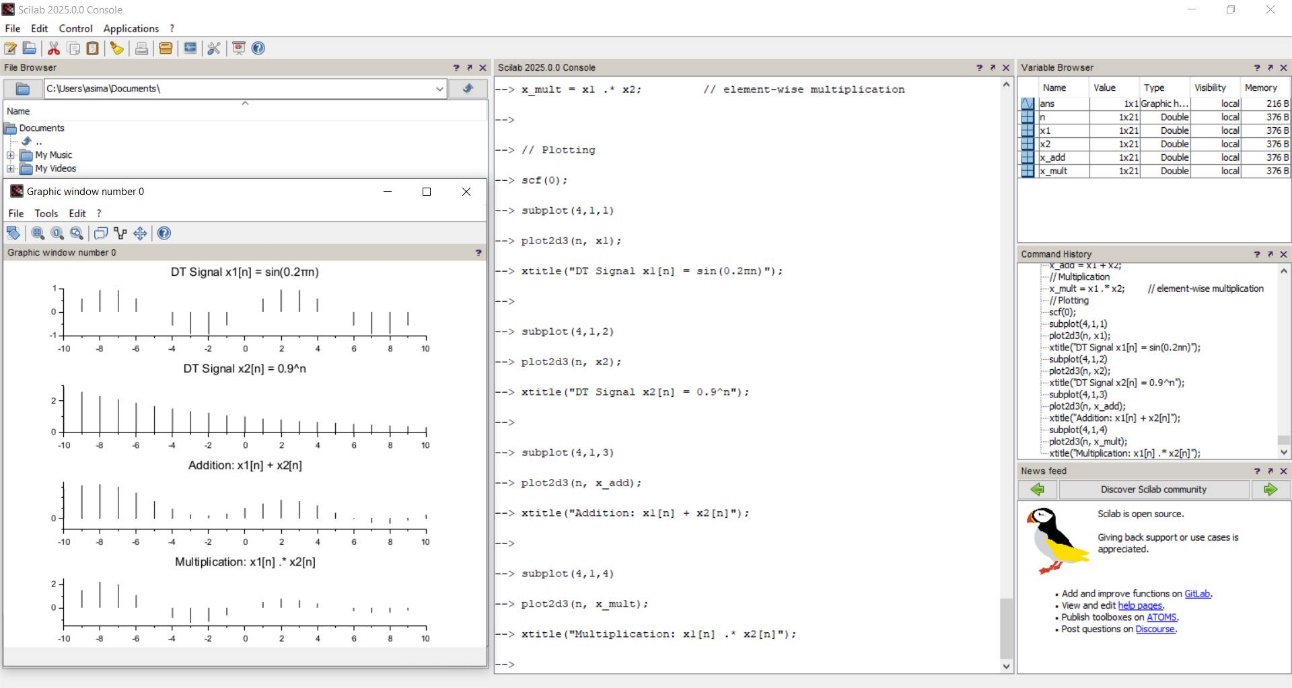
plot2d3(n, x\_add\_dt);

xtitle("DT: x1[n] + x2[n]", "n", "Amplitude");

subplot(4,1,4);

plot2d3(n, x\_mult\_dt);

xtitle("DT: x1[n] \* x2[n]", "n", "Amplitude");



EXERCISE-3

clc;

clear;

close();

// ====================================

// Define 5-Point Discrete-Time Signals

// ====================================

x = [1, 2, 3, 4, 5]; // Input signal x[n]

h = [2, -1, 0, 1, 2]; // Impulse response h[n]

// ====================================

// Perform Linear Convolution

// ====================================

y = conv(x, h); // Linear convolution y[n] = x[n] \* h[n]

// ====================================

// Define Time Indices for Proper Plot Alignment

// ====================================

nx = 0:length(x)-1; // Time indices for x[n]

nh = 0:length(h)-1; // Time indices for h[n]

ny = 0:length(y)-1; // Time indices for y[n]

// ====================================

// Plot the Signals (Discrete using plot2d3)

// ====================================

scf(1); // New figure window

// ---- x[n] ----

subplot(3,1,1);

plot2d3(nx, x);

xtitle("Input Signal x[n] = [1 2 3 4 5]", "n", "Amplitude");

// ---- h[n] ----

subplot(3,1,2);

plot2d3(nh, h);

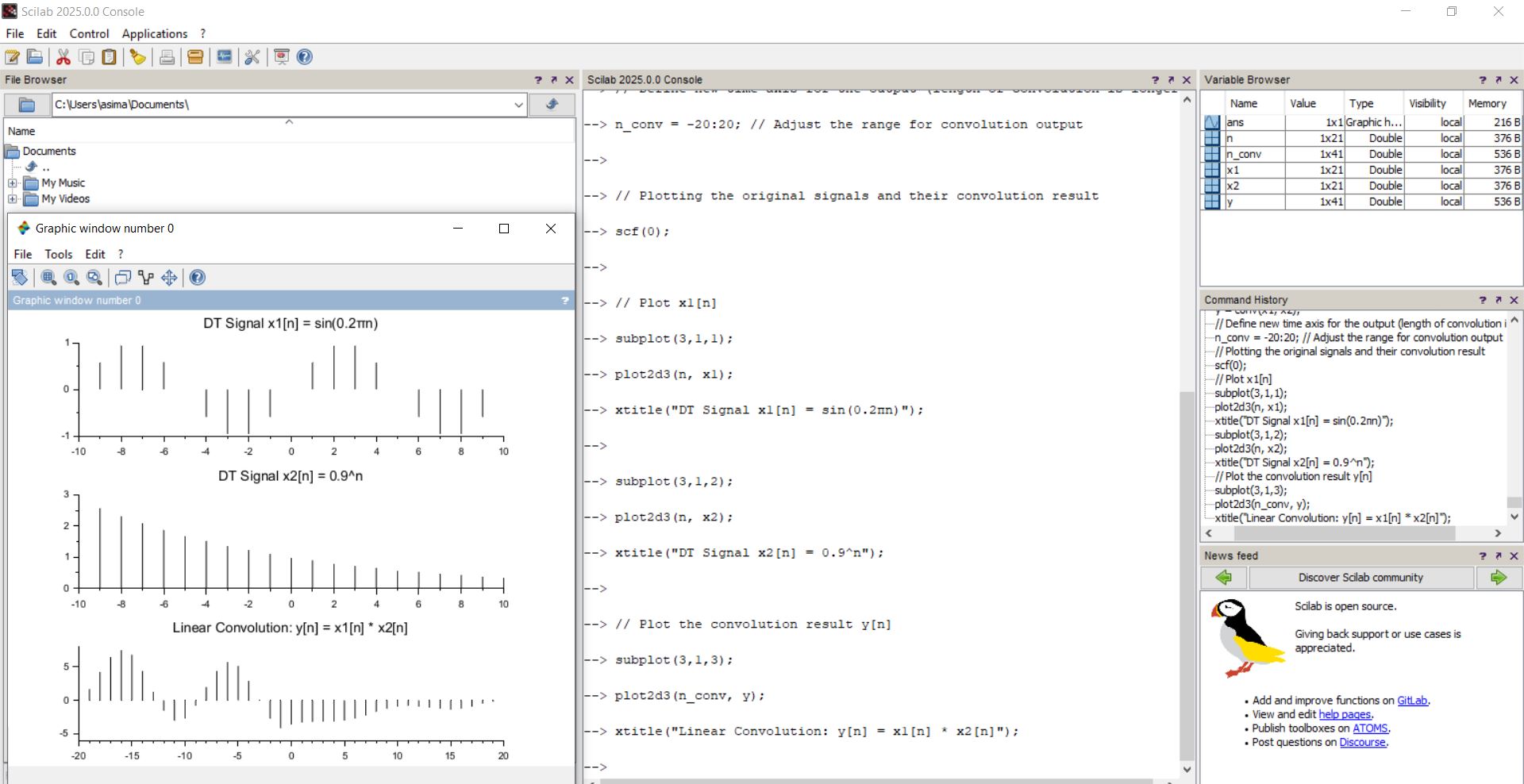
xtitle("Impulse Response h[n] = [2 -1 0 1 2]", "n", "Amplitude");

// ---- y[n] = x[n] \* h[n] ----

subplot(3,1,3);

plot2d3(ny, y);

xtitle("Output y[n] = x[n] \* h[n] (Linear Convolution)", "n", "Amplitude");



Exercise 4: Circular Convolution ( DT Signals)

%% Exercise 4 - Circular Convolution

x = [1, 2, 1, 0];

h = [1, 0, 2, 1];

N = length(x);

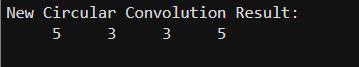
X = fft(x, N);

H = fft(h, N);

y\_circular = ifft(X .\* H);

disp('New Circular Convolution Result:');

disp(y\_circular);



Exercise 5: 4-point and 8-point DFT

%% Exercise 5 - DFT

% 4-point DFT

x4 = [3, 1, 2, 0];

X4 = fft(x4, 4);

disp('4-point DFT (New):');

disp(X4);

% 8-point DFT

x8 = [1, 3, 2, 4, 0, -1, 2, 1];

X8 = fft(x8, 8);

disp('8-point DFT (New):');

disp(X8);

