```
% --- Simulation Parameters (Common to all modulations) ---
clear; clc; close all;

% Carrier properties
carrier_amplitude = 1;      % Amplitude of the carrier wave
base_carrier_frequency = 10; % Hz (Base frequency for all carriers)
sampling_rate = 1000;       % Samples per second (for smooth waveforms)

% Data properties
bit_duration = 1;           % Seconds per bit
binary_data = [1, 0, 1, 1, 0, 1, 0, 0]; % The message to transmit
num_bits = length(binary_data);

% Time vectors
% Corrected t_bit_segment to match Python's linspace(..., endpoint=False)
% This creates a vector from 0 up to (but not including) bit_duration
t_bit_segment = 0 : 1/sampling_rate : bit_duration - 1/sampling_rate;

% Recalculate segment length based on the actual generated segment
t_bit_segment_len = length(t_bit_segment);



total_simulation_time_len = num_bits * t_bit_segment_len;
% The total time vector will now also run from 0 up to (but not including) num_bits * bit_duration
total_simulation_time = 0 : 1/sampling_rate : (num_bits * bit_duration) - 1/sampling_rate;
```

```
% --- Message Signal (For plotting clarity) ---
% Create an NRZ (Non-Return-to-Zero) version for better visualization of binary changes
message_signal_plot = repelem(binary_data, t_bit_segment_len);
% For PSK, it's often useful to represent 0 as -1 and 1 as +1
nrz_data_for_psk = 2 * binary_data - 1;
nrz_signal_plot = repelem(nrz_data_for_psk, t_bit_segment_len);



% --- 1. Amplitude Shift Keying (ASK) Simulation ---
fprintf('\n--- Simulating ASK ---\n');
% Carrier signal (constant for reference)
ask_carrier_signal = carrier_amplitude * sin(2 * pi * base_carrier_frequency *
total_simulation_time);

% ASK modulated signal: '1' means carrier is ON, '0' means carrier is OFF
ask_modulated_signal = zeros(1, total_simulation_time_len);
for i = 1:num_bits
  start_idx = (i - 1) * t_bit_segment_len + 1;
  end_idx = i * t_bit_segment_len;
  if binary_data(i) == 1
    ask_modulated_signal(start_idx:end_idx) = ask_carrier_signal(start_idx:end_idx);
  end
  % If bit is 0, it remains 0 (no signal)
end



% --- 2. Frequency Shift Keying (FSK) Simulation ---
fprintf('--- Simulating FSK ---\n');
```

```
% Define two distinct frequencies for '0' and '1'
fsk_freq_0 = base_carrier_frequency;    % e.g., 10 Hz for '0'
fsk_freq_1 = base_carrier_frequency * 2; % e.g., 20 Hz for '1'

fsk_modulated_signal = zeros(1, total_simulation_time_len);
for i = 1:num_bits
   start_idx = (i - 1) * t_bit_segment_len + 1;
   end_idx = i * t_bit_segment_len;

   if binary_data(i) == 1
      current_frequency = fsk_freq_1;
   else % binary_data(i) == 0
      current_frequency = fsk_freq_0;
   end

   % Generate carrier for this bit segment with the chosen frequency
   segment_carrier = carrier_amplitude * sin(2 * pi * current_frequency * t_bit_segment);
   fsk_modulated_signal(start_idx:end_idx) = segment_carrier;
end


% --- 3. Phase Shift Keying (PSK) Simulation (Binary PSK - BPSK) ---
fprintf('--- Simulating PSK ---\n');
% Carrier signal for PSK
psk_carrier_signal = carrier_amplitude * sin(2 * pi * base_carrier_frequency *
total_simulation_time);

% BPSK modulated signal: phase shift of 180 degrees (multiplying by -1) for '0'
```

```
psk_modulated_signal = zeros(1, total_simulation_time_len);
for i = 1:num_bits
  start_idx = (i - 1) * t_bit_segment_len + 1;
  end_idx = i * t_bit_segment_len;

  if binary_data(i) == 1
    % Phase 0 (or +1 multiplier)
    psk_modulated_signal(start_idx:end_idx) = psk_carrier_signal(start_idx:end_idx);
  else % binary_data(i) == 0
    % Phase 180 degrees (or -1 multiplier)
    psk_modulated_signal(start_idx:end_idx) = -psk_carrier_signal(start_idx:end_idx);
  end
end



% --- Plotting All Waveforms ---
figure('Name', 'Digital Modulation Waveforms');
sgtitle(sprintf('Digital Modulation Waveforms (Data: %s)', num2str(binary_data)),
'FontSize', 16);

% --- ASK Plot ---
subplot(3, 1, 1);
plot(total_simulation_time, message_signal_plot, 'b--', 'LineWidth', 1.5, 'DisplayName',
'Binary Data (for reference)');
hold on;
plot(total_simulation_time, ask_modulated_signal, 'r', 'LineWidth', 1, 'DisplayName',
'ASK Modulated Signal');
hold off;
title('Amplitude Shift Keying (ASK)');
```

```
xlabel('Time (s)');

ylabel('Amplitude');

grid on;

legend('Location', 'northwest');

ylim([-carrier_amplitude*1.1, carrier_amplitude*1.5]); % Adjust y-limit for visibility


% --- FSK Plot ---

subplot(3, 1, 2);

plot(total_simulation_time, message_signal_plot, 'b--', 'LineWidth', 1.5, 'DisplayName',
'Binary Data (for reference)');

hold on;

plot(total_simulation_time, fsk_modulated_signal, 'g', 'LineWidth', 1, 'DisplayName',
'FSK Modulated Signal');

hold off;

title('Frequency Shift Keying (FSK)');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

legend('Location', 'northwest');

ylim([-carrier_amplitude*1.1, carrier_amplitude*1.1]);


% --- PSK Plot ---

subplot(3, 1, 3);

plot(total_simulation_time, nrz_signal_plot, 'b--', 'LineWidth', 1.5, 'DisplayName', 'Binary
Data (NRZ format: 1/-1)');

hold on;

plot(total_simulation_time, psk_modulated_signal, 'm', 'LineWidth', 1, 'DisplayName',
'PSK Modulated Signal');

hold off;
```

```
title('Phase Shift Keying (PSK - BPSK)');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;
legend('Location', 'northwest');
ylim([-carrier_amplitude*1.5, carrier_amplitude*1.5]);

% Adjust layout to prevent title overlap
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]); % Maximize figure
sgtitle(sprintf('Digital Modulation Waveforms (Data: %s)', num2str(binary_data)), 'FontSize', 16);
drawnow; % Ensure plots are rendered

fprintf('\nSimulation complete. Waveforms displayed in MATLAB figure window.\n');
```