1. We find the true solution by using separation of variables and integrating.

\[ \dot{x} = \frac{x^2}{4} \implies \int_2^x \frac{dy}{y^2} = \int_0^t \frac{dt'}{4} \implies -\frac{1}{x} + \frac{1}{2} = \frac{t}{4} \implies \bar{x}(t) = -\frac{1}{-1/2 + t/4}, \quad \bar{x}(3/2) = -\frac{1}{-1/2 + 3/8} = 8 \]

**Euler’s Method:**

Initial time: \( t_0 = 0 \), terminal time \( T = 3/2 \), \( n = 3 \), step-size \( h = (T - t_0)/n = (3/2 - 0)/3 = 1/2 \).

\( t_1 = t_0 + h = 1/2, \quad t_2 = t_1 + h = 1, \quad t_3 = t_2 + h = 3/2 \). By Euler’s method

\[
\begin{align*}
  x(t_1) &= x(t_0) + f(x(t_0))h = 2 + \frac{2^2}{4} \times \frac{1}{2} = 2.5 \\
  x(t_2) &= x(t_1) + f(x(t_1))h = 2.5 + \frac{2.5^2}{5} \times \frac{1}{2} = 3.28125 \\
  x(t_3) &= x(t_2) + f(x(t_2))h = 3.28125 + \frac{3.28125^2}{4} \times \frac{1}{2} = 4.6270752 \ldots
\end{align*}
\]

The error at the terminal time is \( e_T = |8 - 4.6270752 \ldots| = 3.372925 \ldots \)

2.

```matlab
1 % Objective: Compute the Euler approximation of the given ODE
2 function x = Euler(t0, T, n, x0, f)
3 % Inputs:
4 % t0 = Initial time
5 % T = Final time
6 % n = Number of steps
7 % x0 = Initial condition
8 % f = RHS of differential equation
9 % Outputs:
10 % x = Euler approximation of trajectory
11 % Step size
12 h = (T-t0)/n;
13 % Vector to store time steps
14 t = linspace(t0, T, n+1);
15 % Vector to store Euler approximation of trajectory
16 x = zeros(1, n+1);
17 x(1) = x0;
18 for i = 1 : n
19    x(i+1) = x(i) + f(x(i), t(i)) * h;
20 end
21 end
```

```matlab
1 % Objective: Compute the RHS of the ODE in Problem 1
2 function y = X2by4(x, t)
3 % Inputs: x, t
```
% Objective: Compute the Euler solution for different values of n, plot the trajectories and plot the error versus n on a log-log plot
% Error vector
% Values of n
n = [3, 10, 100, 1000, 10000, 100000];
% Plotting approximate trajectories for different values of n
for k = 1 : 6
t = linspace(0, 1.5, n(k)+1);
x = Euler(0, 1.5, n(k), 2, @(x, t)X2by4(x, t));
hold on
plot(t, x)
% Computing the errors
e(k) = abs(x(n(k)+1)-8);
end
legend('$n = 3$', '$n = 10$', '$n = 100$', '$n = 1000$', '$n = 10000$', '$n = 100000$);
xlabel('$t$', 'Interpreter', 'Latex')
ylabel('$x(t)$', 'Interpreter', 'Latex')
title('Plot of trajectories using the Euler method')
hold off
% Log-log plot of errors versus n
figure
plot(log10(n), log10(e), '-o')
xlabel('$\log_{10} n$', 'Interpreter', 'Latex')
ylabel('$\log_{10} e_T$', 'Interpreter', 'Latex')
title('Log-log plot of errors versus n')
The slope of the log-log plot of the error $e_T$ versus $n$ is approximately $-1$. This is consistent with the fact that $e_T(n) = O(n^{-1})$.

3.

% Objective: Compute the RHS of the ODE in Problem 3
function y = Prob3(x, t)
    % Inputs: x, t
    % Outputs: $\sin(t)/2 - \cos(x+t) \cdot \text{atan}(x)$
    y = sin(t)/2 - cos(x+t) * atan(x);
end

% Objective: Compute the Euler solution for different values of n,
% plot the trajectories and plot the error versus n on a log-log plot
% Error vector
e = zeros(7, 1);
% Values of n
n = [3, 10, 100, 1000, 10000, 100000, 1000000];
% Plotting approximate trajectories for different values of n
x = {};
for k = 1 : 7
    t = linspace(0, 2*pi, n(k)+1);
    x{k} = Euler(0, 2*pi, n(k), 2, @(x, t)Prob3(x, t));
    hold on
    plot(t, x{k})
end
for k = 1 : 7
    e(k) = abs(x{k}(n(k)+1)-x{7}(n(7)+1));
end
legend('$n = 3$', '$n = 10$', '$n = 100$', '$n = 1000$', '$n = 10000$', '$n = ... 1000000$',...
' $n = 10000000$','Interpreter', 'Latex')
xlabel('$t$', 'Interpreter', 'Latex')
ylabel('$x(t)$', 'Interpreter', 'Latex')
title('Plot of trajectories using the Euler method')
hold off

% Log-log plot of errors versus n
figure
plot(log10(n(1 : 6)), log10(e(1 : 6)), '-o')
xlabel('$\log_{10}n$', 'Interpreter', 'Latex')
ylabel('$\log_{10}e^{T}$', 'Interpreter', 'Latex')
title('Log-log plot of errors versus n')
The slope of the log-log plot of the error $e_T$ versus $n$ is approximately $-1$. This is consistent with the fact that $e_T(n) = \mathcal{O}(n^{-1})$.

4. The error of the Euler method is $e_T(n) = \mathcal{O}(n^{-1}) \implies e_T(n) \approx C/n$ for some constant $C$. For $n = 100$, $e_T(n) \approx 2 \times 10^{-2}$ so $C \approx e_T(n) \times n \approx 2$. In order to drop the error to $10^{-5}$ we need $C/n \lesssim 10^{-5} \implies n \gtrsim C \times 10^5 = 2 \times 10^5$ (Note that the final time and initial condition are redundant data).

Alternate: We use the fact that $\log e_T(n) = -\log n + C \implies C \approx \log(e_T(n) \times n) = \log(2 \times 10^{-2} \times 100) = \log 2$. In order to drop the error to $10^{-5}$ we need $-\log n + \log 2 \lesssim \log 10^{-5} \implies n \gtrsim 2 \times 10^5$.

5. Here, the domain of interpolation is $[x_0, x_1] = [2, 4]$ and the function is $f(x) = 1/x$. Therefore, the linear interpolation is, $p_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) = \frac{1}{2} + \frac{1}{4 - 1/2}(x - 2) = 3/4 - x/8$. Using Theorem 2.1 in textbook, the error bound can be calculated as follows

$$e(x) = |f(x) - p_1(x)| \leq \frac{1}{8}(x_1 - x_0)^2 \max_{x \in [x_0, x_1]} |f''(\xi)| = \frac{1}{8}(4 - 2)^2 \max_{\xi \in [2, 4]} \frac{2}{\xi^3} = \frac{1}{2} \times \frac{2}{2^3} = \frac{1}{8}$$

since $2/x^3$ is a decreasing function when $x > 0$. Now, to find the actual maximum error we write $e(x) = |g(x)|$ where $g(x) = f(x) - p_1(x) = 1/x - 3/4 + x/8$ and find the local extrema of $g(x)$ in the interval $[2, 4]$. To find the extrema we set $g'(x) = 0 \implies -1/x^2 + 1/8 = 0 \implies x = \pm \sqrt{8}$. We discard $x = -\sqrt{8}$ since it does not lie in the domain of interpolation $[2, 4]$. We note that $g''(\sqrt{8}) > 0$ and so $g(x)$ has a local minimum at $x = \sqrt{8} \implies e(x) = |g(x)|$ has a local maximum at $x = \sqrt{8}$ (this can also be verified by looking at the plot below).

The actual maximum error is $e(\sqrt{8}) = |1/\sqrt{8} - 3/4 + \sqrt{8}/8| = |2/\sqrt{8} - 3/4| = 0.0429\ldots < 1/8$.

6. The interval is $[a, b] = [1, 4]$. Let the number of pieces that minimizes the maximum error be $n$ and the length of each piece be $h = (b - a)/n$. Using Theorem 2.1 from the textbook we have $e \leq \frac{1}{18}h^2 \max_{\xi \in [1, 4]} |f''(\xi)|$. Now $f(x) = x \ln x$, $f'(x) = \ln x + 1$, $f''(x) = 1/x$ which is monotonically decreasing in $[1, 4] \implies \max_{\xi \in [1, 4]} |f''(\xi)| = 1$. Therefore, $e \leq h^2/8 < 10^{-3} \implies h = (4 - 1)/n < \sqrt{8} \times 10^{-3} = 0.08944\ldots \implies n > 3/0.08944\ldots = 33.5420\ldots$. Since $n$ is a positive integer we may take $n = 34$. 