1 (20 pts). (a) Write a MATLAB function \texttt{Newton(f,df,x,tol,miter)} that implements Newton’s method. Write separate files \texttt{f.m} and \texttt{df.m} that contain \( f \) and its derivative \( f' \), respectively. The input parameters are the initial guess \( x \), the desired error tolerance \( tol \), and the maximum number of iterations \( miter \). The program should stop when either two consecutive iterates satisfy
\[
\text{abs}(x_{k+1} - x_k) \leq tol \quad \text{or} \quad miter \quad \text{is reached.}
\]
The output parameters are the approximate zero, an error estimate, and the number of iterations.

(b) Find the smallest positive root of the equation
\[
e^{-x} = \sin(x)
\]
with \( tol=10^{-15} \) and \( miter=50 \).

(c) Find the zeros of
\[
f(x) = x^3 - x^2 - x + 1
\]
using \texttt{Newton}. Plot \( f \) with \texttt{fplot} to figure out reasonable starting points. Modify the code so that it can determine the order of convergence. Provide a theoretical justification to your findings.

2 (20 pts). (a) Write a MATLAB function \texttt{Secant(f,x0,x1,tol,miter)} that implements the secant method with starting values \( x_0 \) and \( x_1 \). The other input and output variables are the same as in Pb 1.

(b) Solve Pb 1(b) with \( tol=10^{-15} \) and \( miter=50 \). Compare the results and draw conclusions.

3 (20 pts). (a) Given \( f(x) = e^{-x} - \sin x \), determine an interval \([a, b]\) and a contraction \( F : [a, b] \rightarrow [a, b] \) such that the smallest positive zero of \( f \) is a fixed point of \( F \) on \([a, b]\). Write the corresponding functional iteration and study the rate of convergence.

(b) Write a simple MATLAB function that implements this iteration, and computes the number of iterations for an error tolerance of \( 10^{-15} \). Compare with Pbs. 1 and 2 and draw conclusions.

4 (10 pts). Consider the following recursion
\[
x_{k+1} = 2x_k - 3 + \frac{2}{x_k}
\]
(a) Determine the fixed points by hand;
(b) Examine whether the iterations converge locally, and if so find the rate of convergence.

5 (30 pts). (a) Write a simple MATLAB program \texttt{System(f,g,fx,fy,gx,gy,x0,y0,tol,miter)} to solve a 2-by-2 nonlinear system of equations \( f(x, y) = g(x, y) = 0 \) by the Newton’s Method. The input parameters are the functions \( f \) and \( g \) and its partial derivatives, the starting point \((x_0, y_0)\), an absolute error tolerance \( tol \), and the maximum number of iterations \( miter \). For the solution of the resulting linear systems use the MATLAB command ‘\( \setminus \)’.

(b) The following system has four zeros in the domain \((-4, 4) \times (-4, 4)\)
\[
f(x, y) = x^2 + xy^3 - 9 = 0, \quad g(x, y) = 3x^2y - y^3 - 4 = 0,
\]
Use the command \texttt{contour} to plot the zero level sets of both \( f \) and \( g \) in the same picture to determine reasonable initial guesses (use \texttt{help contour} to find out information about the command). Use the commands \texttt{hold on} and \texttt{hold off} to produce this picture. You may also want to zoom the picture with \texttt{zoom}. Use a syntax such as:
\[
x = -4:0.1:4;
y = -4:0.1:4;
[X,Y] = meshgrid(x, y);
contour(x, y, X.^2*X.*Y.^3-9, [0 0])
Figure out what each line does. Note the use of “.” to force elementwise operations.

(c) Use `System` to approximate the four zeros with `tol=10^{-10}`. Determine the number of iterations and the rate of convergence.

(d) Set the function \( h(x, y) = f(x, y)^2 + g(x, y)^2 \). Then the zeros of the system correspond to minima of \( h \). Use the MATLAB built-in minimization function `fmins` to approximate the four zeros. Type `help fmins` and `help foptions` to find out more about `fmins` and its options. In particular, use `options(1)=1` and `options(3)=1.e-10`. 