Goal: Nonlinear equations.

**Problem 1** Solve Problem 6, p.326 from the book.

**Problem 2** Write your own Newton and Chord solvers for single equations. Apply those codes to the resolution of the following equations with the given initial iterate

1. \( F(x) = \cos(x) - x, \ x_0 = 0.5 \),
2. \( F(x) = \arctan(x), \ x_0 = 10 \) and \( x_0 = 1 \),
3. \( F(x) = \sin(x), \ x_0 = 3 \),
4. \( F(x) = x^2, \ x_0 = 0.5 \).

Tabulate or plot the iteration statistics and explain your results.

**Problem 3** 1. Write your own Newton routine for systems (use LU factorization) or alternatively, download one from the web (Prof. Kelley’s page may be useful here). If using your code fully comment it and include the code in the assignment; if using somebody else’s code, make sure you understand everything about it, and provide a commented version with your assignment

2. Consider the system
   \[
   \exp(x_1^2 + x_2^2) - 1 = 0, \\
   \exp(x_1^2 - x_2^2) - 1 = 0.
   \]
   Give all the exact solution(s).

3. Apply your code to the above system. Try the staring points \((0.1, 1)\), \((10, 10)\) and \((20, 20)\). Comment.

**Problem 4** 1. If the magnitude of the Newton step is much larger than the iterate itself, problems can happen. Identify such cases from Problem 2.

2. Discuss, explain and implement the Armijo rule which in the scalar case takes the form
   \[
   \begin{align*}
   r_0 &= |f(x)|, \\
   \text{do while } |f(x)| &> \tau_r r_0 + \tau_a \\
   &\quad \text{if } f'(x) = 0 \text{ terminate with failure} \\
   &\quad \text{if } s = -f(x)/f'(x) \text{ (search direction)} \\
   &\quad \lambda = 1 \\
   &\quad \text{do while step not accepted}
   \end{align*}
   \]
- $x_t = x + \lambda s$ (trial point)
- if $|f(x_t)| < (1 - \alpha \lambda)|f(x)|$ then
  - $x = x_t$ (accept step)
  - else
  - $\lambda = \lambda/2$ (reject step)
- end
- end

The parameter $\alpha$ is usually taken small ($\alpha = 10^{-4}$).

3. Propose a general (non scalar) version of the above algorithm.

4. Repeat Problem 2, point 2. Compare the performance of your code with the generic Newton's code available from Tim Kelley's webpage

{http://www4.ncsu.edu/~ctk/newtony.html}

and with the \texttt{fsolve} routine from MATLAB. Comment.