1. Verify analytically that Steffensen's method (with a sufficiently good initial guess \( x_0 \)) in fact is quadratically convergent when applied to the fixed point iteration \( x_{n+1} = g(x_n) \).

2. The nonlinear system
\[
\begin{align*}
  x^2 - y - 1 &= 0 \\
  (x - 2)^2 + (y - \frac{1}{2})^2 - 1 &= 0
\end{align*}
\]
has two real solutions.
   
a. Implement Newton's method for systems, and apply it to this system of equations, starting with \((0,0)\) and with \((2,2)\). Print out the successive iterates to quite high precision, so that you can see how convergence proceeds.

   b. Choose some complex starting point, and see if you can catch one of its two complex solutions (you'll then find the second complex solution by conjugating both \(x\) and \(y\)).

3. Consider the system
\[
\begin{align*}
  x &= \frac{1}{\sqrt{2}} \sqrt{1 + (x+y)^2} - \frac{2}{3} \\
  y &= \frac{1}{\sqrt{2}} \sqrt{1 + (x-y)^2} - \frac{2}{3}
\end{align*}
\]
Find a region \(D\) in the \(x,y\)-plane for which a fixed point iteration
\[
\begin{align*}
  x_{n+1} &= \frac{1}{\sqrt{2}} \sqrt{1 + (x_n+y_n)^2} - \frac{2}{3} \\
  y_{n+1} &= \frac{1}{\sqrt{2}} \sqrt{1 + (x_n-y_n)^2} - \frac{2}{3}
\end{align*}
\]
is guaranteed to converge to a unique solution for any \((x_0,y_0) \in D\).
   
a. State clearly what properties this region must have.

   b. Find a region with these properties and show that it has these properties.

4. Apply the Sturm sequence technique to check how many real zeros the polynomial
\(x^4 - x^3 - 2x^2 - 2x + 4\) has in the interval \([-2, 2]\), and also on the full real axis.

Next page describes how the Sturm sequence technique is applied.
The following is an example that illustrates how one uses the Sturm’s sequence method to locate polynomial roots along the real line:

**Example:** Let \( f(x) = x^6 + 4x^5 + 4x^4 - x^2 - 4x - 4 \) \( = (x^2 + 1)(x^2 - 1)(x + 2)^2 \).

Consider then the following sequence of polynomials of successively decreasing orders:

\[
\begin{align*}
f_1 &= f(x) = x^6 + 4x^5 + 4x^4 - x^2 - 4x - 4 \\
f_2 &= f' = 6x^5 + 20x^4 + 16x^3 - 2x - 4 \\
f_3 &= \begin{cases} 
eg & \text{negative of remainder} \\
& \text{in long division } f_1/f_2 \\
& \text{(scale to get 'nice'} \\
& \text{integer coefficients if} \\
& \text{so desired)} \\
f_4 &= \begin{cases} \text{same as above, but} \\
& \text{based on } f_2/f_3 \\
& \text{so desired} \end{cases} = 4x^4 + 8x^3 + 3x^2 + 14x + 16 \\
f_5 &= \{ \text{... based on } f_3/f_4 \} = -17x^2 - 58x - 48 \\
f_6 &= \{ \text{... based on } f_4/f_5 \} = -x - 2 \\
f_7 &= \{ \text{... based on } f_5/f_6 \} = 0
\end{align*}
\]

Next, evaluate the signs of these polynomials at some set of points, for ex. at \( x = -\infty, +\infty, 0, -\frac{24}{17} \). We get

<table>
<thead>
<tr>
<th></th>
<th>-\infty</th>
<th>+\infty</th>
<th>0</th>
<th>-\frac{24}{17}</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

where the bottom line shows the number of sign changes there is in each column. The difference between any of these numbers tells how many distinct zeros (i.e. ignoring multiplicity) \( f(x) \) has in the corresponding interval. In the present example, there are thus 3 distinct real roots altogether (between +\( \infty \) and -\( \infty \)), one of which is seen to be located between 0 and \( \infty \), etc.

The procedure can be implemented nicely in both Mathematica and Matlab:

**Mathematica:**

\[
\begin{align*}
f_1 &= x^6 + 4x^5 + 4x^4 - x^2 - 4x - 4 \\
f_2 &= D[f_1, x] \\
f_3 &= \text{PolynomialRemainder}[f_1, f_2, x] \\
&\quad \text{...}
\end{align*}
\]

**Matlab:**

\[
\begin{align*}
f1 &= [1 4 4 0 -1 -4 -4]; \\
f2 &= \text{polyder}(f1); \\
[~,f3] &= \text{deconv}(-f1,f2); \\
f3(1:find(f3,1,'first')-1) &= [ ]; \\
&\quad \text{...}
\end{align*}
\]