Fixed point iterations and solution of non-linear functions

\[ f(x) = 0 \]
Zero of a Nonlinear Function

- **Problem definition:**
  - Find the solution of the equation \( f(x) = 0 \)
  - for scalar valued \( f \) and \( x \);
  - Look for the solution either in
    - An interval, generally \(-\infty < x < \infty\)
    - In the uncertainty interval \([a, b]\), where \( f(a)f(b) < 0 \)

- **Types of algorithms available:**
  1. Bisection method
  2. Substitution methods
  3. Methods based on function approximation

- **Assumptions:**
  - In the defined intervals, **at least one** solution exists
  - We are looking for **one solution**, **not all** of them
Fixed point iterations

- Fixed point iteration formulas generally have the form
  \[ x_{k+1} = F(x_k) \]

- A fixed point of \( F \) is a point \( x^* \), where
  \[ x^* = F(x^*) \]

- A fixed point iteration is called consistent with a non-linear equation \( f(x) \), if
  \[ f(x^*) = 0 \iff x^* = F(x^*) \]
Convergence order of fixed point iterations

- For any (converging) fixed point iteration, we can write
  \[ |x_{k+1} - x_k| \leq c \cdot |x_k - x_{k-1}|^q \]

- where \(c\) is the rate of convergence and \(q\) is the convergence order

- If we take the logarithm on both sides, we get
  \[ \log(|\Delta x_{k+1}|) = q \cdot \log(|\Delta x_k|) + \log(c) \]

- Which we can use to fit an average \(q\) and \(c\)
  \[ Y = a \cdot X + b \]
Bisection Method

1. Define starting interval \([a, b]\) (check that \(f(a) \times f(b) < 0\))

2. Compute \(x = \text{mean}(a, b)\)

3. Redefine the interval
   - Set either \(a = x\) or \(b = x\) so that \(f(a) \times f(b) < 0\) is still fulfilled

4. Iterate 2 and 3 until the requested precision is reached

- **Advantages**
  - After \(n\) iterations, the interval is reduced by \(2^n\)
  - Final precision can be predicted a priori

- **Disadvantages**
  - Function characteristics are not used to speed up the algorithm
Newton Method

- The Newton method is based on Taylor expansion
  \[
  f(x) \approx f(x_0) + f'(x_0)(x - x_0) + O(\Delta x^2)
  \]
  \[
  f(x) \to 0 \implies x \approx x_0 - \frac{f(x_0)}{f'(x_0)} \implies x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}
  \]

- Advantages
  - It has 2\textsuperscript{nd} order convergence
  \[
  \left| \varepsilon_{k+1} \right| = c \left| \varepsilon_k \right|^2
  \]

- Disadvantages
  - Convergence is not guaranteed even if the uncertainty interval is known
  - If the derivative must be calculated numerically, the secant method is more convenient

\[
\begin{align*}
f(x) &= x^2 - 1 \\
f'(x) &= 2x
\end{align*}
\]
Secant Method

- The Secant is based on the same principles as the Newton method, but it approximates the derivative numerically

\[ x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \]

\[ f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \]

- **Advantages**
  - Does not require the analytical first order derivative

- **Disadvantages**
  - Convergence is not assured even if the uncertainty interval is known
  - Convergence order is only 1.618

\[ x_0 = 1.8 \]
\[ x_1 = 1.7 \]
How does Matlab do it? Nonlinear Functions

- **fzero** finds the zero of a scalar valued function; It uses a combination of bisection, secant, and inverse quadratic interpolation methods.

- **roots** finds all the roots of a polynomial function; It computes the eigenvalues of the companion matrix, which correspond to the roots of the polynomial.

```
A = diag(ones(n-1,1),-1);
A(1,:) = -c(2:n+1)./c(1);
eig(A);
```

- Where `c` is the vector of the polynomial coefficients

\[ p = c_1 x^n + c_2 x^{n-1} + \ldots + c_n x^1 + c_{n+1} \]
Matlab Syntax Hints

- \( x = \texttt{fzero} \left( \texttt{fun}, \ x0 \right); \)
  - \texttt{fun} is a function taking as input a scalar \( x \), returning as output the scalar function value \( f(x) \)
  - \( x0 \) is either an initial guess (if it has length 1) or an uncertainty interval (if it has length 2, then \( f(x0(1)) \cdot f(x0(2)) < 0 \) must be fulfilled)

- \( x = \texttt{roots} \left( c \right); \)
  - \( c \) is a vector containing the polynomial coefficients in the order
    \[
p = c_1 x^n + c_2 x^{n-1} + \ldots + c_n x^1 + c_{n+1}
    \]
Exercise 1: Fixed point iterations

- Consider the function

\[
(1) \quad f(x) = x \cdot \exp(x) - 1 = 0
\]

- And the following iterative formulas:

\[
(2) \quad x_{k+1} = \exp(-x_k)
\]

\[
(3) \quad x_{k+1} = \frac{x_k^2 \exp(x_k) + 1}{\exp(x_k)(1 + x_k)}
\]

\[
(4) \quad x_{k+1} = x_k + 1 - x_k \exp(x_k)
\]
Assignment 1

1. For each of the iterative formulas try to find a fixed point
   - Define a starting guess $x_0$ between 0 and 1
   - while $\text{abs}(x_k - x_{k-1}) > 1e-8$, calculate the next $x$ value
   - Store the values calculated in a vector $xvec$
   - Also terminate the while-loop if 1e5 iterations are exceeded

2. Is there a fixed point and is it consistent with (1)?

3. Estimate the convergence orders and the rates of convergence for the formulas which have a fixed point
   - Calculate $dX = \log(\text{abs}(\text{diff}(xvec)))$;
   - Set $X = dX(1:end-1); \text{ and } Y = dX(2:end)$;
   - Perform a linear fit with $p = \text{polyfit}(X,Y,1)$;
   - Calculate $q$ and $c$ from the fitting coefficients in $p$, remember that
     \[
     \log(\|\Delta x_{k+1}\|) = q \cdot \log(\|\Delta x_k\|) + \log(c)
     \]
Exercise: CSTR Multiple Steady States

- Consider a CSTR where a reaction takes place

\[
\begin{align*}
    c_A^{in}, & \quad Q^{in}, \quad T^{in} \\
    & \quad T^C \quad c_i, \quad Q^{out}, \quad T
\end{align*}
\]

- We assume the following
  - \( V = \text{const.} \), i.e. \( Q^{in} = Q^{out} = \text{const.} \)
  - Perfect coolant behavior, i.e. \( T^{C,in} = T^{C,out} = \text{const.} \)
  - Constant density and heat capacity of the reaction mixture
  - Constant reaction enthalpy
  - Constant feed, i.e. \( c_A^{in} = \text{const.}, \ T^{in} = \text{const.} \)

- The reaction rate is given by
  \[
  k(T) = k_0 \exp\left(-\frac{E_A}{RT}\right)
  \]
CSTR Mass and Energy Balances

- The mass and energy balances read

\[ V \frac{dc_A}{dt} = -Vc_A k(T) + Qc_{A}^{in} - Qc_A \]

\[ V \frac{dc_B}{dt} = Vc_A k(T) - Qc_B \]

\[ \dot{Q}^{Total} = \dot{Q}^{Reaction} + \dot{Q}^{In} + \dot{Q}^{Out} + \dot{Q}^{Cool} \]

\[ V \rho c_p \frac{dT}{dt} = Vc_A k(T)(-\Delta_R H) + Q \rho c_p (T^{in} - T) + K^{Cool} (T^{Cool} - T) \]

- With the T-dependent reaction rate constant

\[ k(T) = k_0 \exp\left(-\frac{E_A}{RT}\right) \]
Dimensionless Mass and Energy Balances

- If we define

\[ u_i = \frac{c_i}{c_A^{in}} \]
\[ \theta = \frac{T}{T^{in}} \]
\[ \tau = \frac{Q}{V} t \]

\[ \kappa = \frac{V}{Q} \]
\[ K^C = \frac{K^{Cool}}{Q \rho c_p} \]
\[ \theta^C = \frac{T^{Cool}}{T^{in}} \]

\[ \eta = \frac{(-\Delta_R H) c_A^{in}}{\rho c_p T^{in}} \]
\[ \alpha = \frac{E_A}{RT^{in}} \]

- We get a dimensionless form

\[ \frac{d u_A}{d \tau} = 1 - u_A \left( 1 + \kappa(\theta) \right) \]
\[ \frac{d u_B}{d \tau} = u_A \kappa(\theta) - u_B \]
\[ \frac{d \theta}{d \tau} = \eta u_A \kappa(\theta) + 1 - \theta + K^C \left( \theta^C - \theta \right) \]

\[ u_{A,0} = u_{B,0} = 0 \]
\[ \theta_0 = 1 \]
\[ \kappa(\theta) = \kappa_0 \exp \left( -\frac{\alpha}{\theta} \right) \]
CSTR Temperature Equilibrium

- The steady state concentration of A reads
  \[ u_A^{ss} = \frac{1}{(1 + \kappa(\theta))} \]

- The temperature in steady state is therefore given by
  \[
  \dot{Q}_{\text{Reaction}} + \dot{Q}_\text{In} = -\left( \dot{Q}_\text{Out} + \dot{Q}_\text{Cool} \right) \\
  \kappa(\theta)u_A^{ss} \eta + 1 = \theta - K^C \left( \theta^C - \theta \right) \\
  \frac{\eta \kappa(\theta)}{1 + \kappa(\theta)} + 1 - \theta + K^C \left( \theta^C - \theta \right) = 0 \\
  \kappa(\theta) = \kappa_0 \exp\left(-\frac{\alpha}{\theta}\right)
  \]
Assignment 2

1. Plot the total heat flow from and to the reactor vs. the dimensionless reactor temperature for $\theta = 0.9 \ldots 1.25$
   - Use $\alpha = 49.46; \kappa_0 = 2.17e20; K_C = 0.83; \eta = 0.33$ and $\theta_C = 0.9$.

2. Implement and use the secant method in a function to find the three steady state temperature of the CSTR.
   - Use a function of the form
     ```
     function [x, xvec] = secantRoot(f, x0)
     ```
   - Also return the x-values calculated as a vector `xvec`.
   - The calculation steps of the secant method can be found on slide 7
   - The secant method uses two starting guesses; from `x0`, calculate $x1 = (1+\varepsilon)x0$. Suggest a value for $\varepsilon$ (not too small, why?).
   - Loop while $\text{abs}(x_k - x_{k-1}) > 1e-8$ and $f(x_k) > 1e-6$ and $n < 1e5$
   - You will have to work with three x-values at any given iteration, that is $x_{k+1}, x_k$ and $x_{k-1}$
Assignment 2 (continued)

3. In what range of $x_0$ can you converge to the intermediate solution? What feature of the function determines which solution is found?

4. Use one of the resulting $x_{vec}$ to estimate the convergence order and rate of convergence of the secant method.