Computer Methods in Engineerings
Exercise on the bracket method

Problem 1

Bracket search methods are described in the notes.

a) Implement the bracket search method to locate a minimum of the function $f(x) = x^2 - sin(x)$ on the interval $(a, b) = (0, 1)$. Split the interval (a, b) into four subintervals of length $\Delta x = (b - a)/4.0$ and initially select α and β by $\alpha = a + \Delta x, \beta = b - \Delta x$. Experiment with different values of Δx.

b) Implement the golden ration method to locate a minimum of the function $f(x) = x^2 - sin(x)$ on the interval $(a, b) = (0, 1)$. Compare the number of iterations used to find a solution with the number of iterations used in the previous point.

Problem 2

Assume you are going to build a pipeline to transport CO2 from a capture site to an offshore storage site. The pipeline has a length of 1000 km, and the flow rate of CO2 is 0.3 m^3/s. The viscosity of CO2 is approximately 15 Pa s at room temperature. The cost of the pipe is 10 dollar per m^2 of pipe surface, and the cost of operating the pipeline is 1E-11 dollar per pascal per m^3 of gas transported. The income from selling the transport service is 1 dollar per m^3 of gas transported. Here, we disregard the expansion of the gas due to pressure changes, and assume that the flow is laminar. We will thus use the Hagen-Poiseulle equation to calculate the pressure drop in the pipe. A code for calculating the profit from the pipeline as a function of the radius of the pipe is given as follows:

```
import numpy as np
import math as m
import matplotlib.pyplot as plt


class CpipedGas:
    def __init__(self):
        self.pricePerPascalperCubed=1E-11 #dollar per pascal per cube of gas
        self.materialPrice=10 #dollar per m2
        self.flowRate=0.3 #m3 per second
        self.length=1E6 #1000 km
        self.viscosity= 15 # Pa s, approximately viscosity of CO2 at room ...
            temperature
        self.payment=1 # dollar per cube

    def costPipe(self,radius):
        return m.pi*radius**2*self.length*self.materialPrice
```

```
    def costUse(self,radius):
        # Using Hagen-Poiseulle to calculate pressure drop
        deltaP=8*self.viscosity*self.length*self.flowRate/(m.pi*radius**4)
        return deltaP*self.pricePerPascalperCubed

    def profit(self,radius,time):
        return self.payment*self.flowRate*time - ...
            self.costUse(radius)*self.flowRate*time - self.costPipe(radius)
```

Use this code to find the optimal radius of the pipe using the bracket method, when we assume that we will operate the pipe for 1E8 seconds (a bit more than 3 years).