Computational Methods and Modelling

Antonio Attili & Edward McCarthy

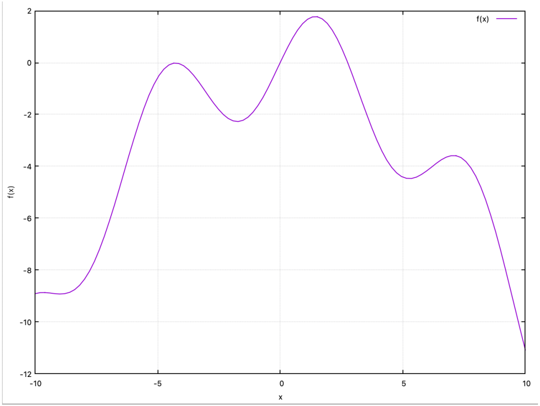
antonio.attili@ed.ac.uk
ed.mccarthy@ed.ac.uk

School of Engineering
University of Edinburgh
United Kingdom

Solution tutorial 11
Optimisation.

Exercise 1: Optimisation using Golden Search

- It is always a good idea to plot the function first, to identify the range that contains the global maximum. This helps in the initialisation of the Golden Search method.



Exercise 1: Optimisation using Golden Search

- The python code is the following

”

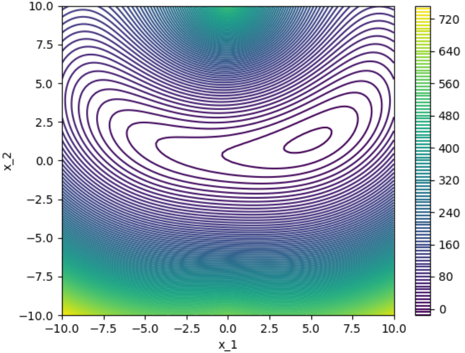
```
import numpy as np
import math as math

def gsection(ftn, xl, xm, xr, tol = 1e-9):
    gr1 = 1 + (1 + np.sqrt(5))/2
    # successively refine x.l, x.r, and x.m
    fl = ftn(xl)
    fr = ftn(xr)
    fm = ftn(xm)
    while ((xr - xl) > tol):
        if ((xr - xm) > (xm - xl)):
            y = xm + (xr - xm)/gr1
            fy = ftn(y)
            if (fy >= fm):
                xl = xm
                fl = fm
                xm = y
                fm = fy
            else:
                xr = y
                fr = fy
        else:
            y = xm - (xm - xl)/gr1
            fy = ftn(y)
            if (fy >= fm):
                xr = xm
                fr = fm
                xm = y
                fm = fy
            else:
                xl = y
                fl = fy
    print(ftn(xm))
    return(xm)

xl=0
xm=5
xr=10
def ftn(x):
    return 2*math.sin(x)-(x**2/10)
print(gsection(ftn, xl, xm, xr, tol = 1e-9))
```

Exercise 2: Optimization of spring system

- We need to minimise the potential energy $PE(x_1, x_2)$, which is a function of the two variables x_1 and x_2.
- A good technique to achieve this is the Newton method for two variables, which we can run using the standard package `scipy.optimize`.
- It is convenient to plot the function $PE(x_1, x_2)$. For example, we can plot contours of the function. We can see that the minimum is located around the position $(x_1, x_2) = (5, 2)$.



Exercise 2: Optimization of spring system

- The plot can be done with the following code.

''

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

x1_a = np.linspace(-10,10,100)
x2_a = np.linspace(-10,10,100)

x1, x2 = np.meshgrid(x1_a, x2_a, indexing='ij')

ka=9.
kb=2.
La=10.
Lb=10.
F1=2.
F2=4.
PE = 0.5*(ka*((x1**2+(La-x2)**2)**0.5 - La)**2)+0.5*\
    (kb*((x1**2+(Lb+x2)**2)**0.5 - Lb)**2)-F1*x1-F2*x2

plt.figure()
plt.contour(x1,x2,PE,100)
plt.colorbar()
plt.xlabel('x_1')
plt.ylabel('x_2')
plt.savefig('PE.png')

plt.show()
```