Approach

Expected values are calculated by averaging 10000 simulated value for X and max(X-100, 0). For dE[max(X-100, 0)]/dsigma, method of forward differnce for partial derivative is used.

In [1]:

```
a = 0.8;
theta = 100;
sigma = 0.2;
X0 = 100;

import numpy as np
import math
#Resolution of discretisation
res = 100
T = 2.0;
dt = T/res;

#No of simulations
max_iter = 10000;

#Initialising
Exp1 = np.zeros(max_iter);
Exp2 = np.zeros(max_iter);
Exp3 = np.zeros(max_iter);

epsilon = sigma*0.0001;
#for every simulation
for j in range(max_iter):
    #draw from random normal variable
    n = np.random.normal(0, 1, res)
    #Initialise X and Xs
    X = X0;
    Xs = X0;
    #for every time interval calculate X ans
    for i in range(res):
        X = X+a*(theta-X)*dt + sigma*n[i]*math.sqrt(X*dt)
        Xs = Xs+a*(theta-Xs)*dt + (sigma+epsilon)*n[i]*math.sqrt(Xs*dt)

    Exp1[j]=X;
    Exp2[j]=max(X-100, 0);
    Exp3[j]=max(Xs-100, 0);

l = [np.mean(Exp1),np.mean(Exp2),(np.mean(Exp3)-np.mean(Exp2))/epsilon];

print(l)
```

```
[100.02430909777132, 0.6300704380138868, 3.1502768030156054]
```