In [22]:
```
from random import uniform, randint
import math

def goal(x):
        return math.sin(1/x)/x

class State:
        def __init__(self, x):
                self.x = x
                self.f = goal(x)

def succ(s):
        x1 = s.x + uniform(-1.0, 1.0)
        return State(x1)

def initialState():
        return State(uniform(0.01,1))

def choose(set):
        x = set[randint(0, len(set)-1)]
        y = set[randint(0, len(set)-1)]
        if(x.f > y.f):
                return x
        return y

def cross(x,y):
        next_x = (x.x + y.x)/2.0
        return State(next_x)

def mutate(s):
        if(uniform(0.0,1.0)<0.8):
                return succ(s)
        return s

population_size = 100
number_of_generations = 100
current_gen = []

for i in range(population_size):
        current_gen.append(initialState())

for i in range(number_of_generations):
        next_gen = []
        for j in range(population_size):
                x = choose(current_gen)
                y = choose(current_gen)
                child = cross(x,y)
                child = mutate(child)
                next_gen.append(child)
        current_gen = next_gen

best = current_gen[0]
for s in current_gen:
        if(s.f > best.f):
                best = s
```

```
print('x = ',best.x, 'f = ',best.f)
```

```
x =  -0.0025595329379093368 f =  354.80237697838896
```