Numerical minimisation

It is often necessary to find the global minima of a function to optimise an analysis. When the function is unknown and may be dependent on several input arguments, numerical minimisation is used.

1. Create a file called `testFunction.py` that contains the Python code that is given in Listing 1. Create another file called `minimise.py` that contains the Python code that is given in Listing 2. These two Python files must be in the same directory. Then run `minimise.py` from the directory that contains the two files.

   Listing 1: testFunction.py

   ```
   import math

   def testFunction(x):
       if x < -2 or x > 5:
           return 0
       return math.cos(x)*(x**2 - 2*x - 2.5) + (-0.4*x + 2.5)
   ```

   Listing 2: minimise.py

   ```
   import testFunction
   from testFunction import testFunction

   x = 1.5
   y = testFunction(x)
   print("testFunction(" + str(x) + ") = " + str(y))
   ```

2. Edit `minimise.py` and write a function that calls `testFunction` with x values between $-2 < x < 5$, in steps of 0.01. Add logic to this function, such that it returns a list of x values for minima that have been found.

   A generic curve $f(x)$ is illustrated in Figure 1. This curve has two minima. The new function within `minimise.py` should return the x-axis position of both of these minima.

   A minima can be found by computing three points: a current point, a previous point and a point before the previous point. These three points are illustrated in Figure 2. In this illustration, the previous point has been identified as a local minimum.

3. Edit `minimise.py` and add a function that:

   - Picks three random x values within the range $-2 < x < 5$.
   - Selects the two lowest values of the three calculated.
   - Selects a new third value at a random x point between the two remaining values.
   - Continues until the difference between the three points or two remaining values is less than 0.001. The difference should be computed as the difference between the points along the x-axis, which is illustrated in Figure 3.
   - Returns the value of x for the resulting minima.

   The selection of three random points is illustrated in Figure 1. A random value within $-2 \leq x \leq 5$ can be generated using the function that is given in Listing 3. Listing 4 demonstrates how to find the index of a list element with the maximum value within the list. Listing 5 demonstrates how to find the index of a list element with the minimum value within the list.

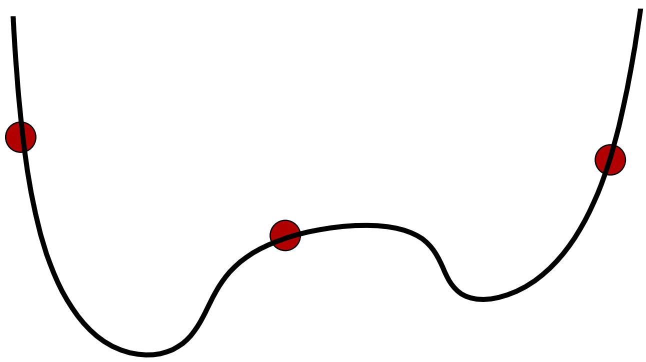

Figure 1: A generic function $f(x)$, where three points have been calculated.

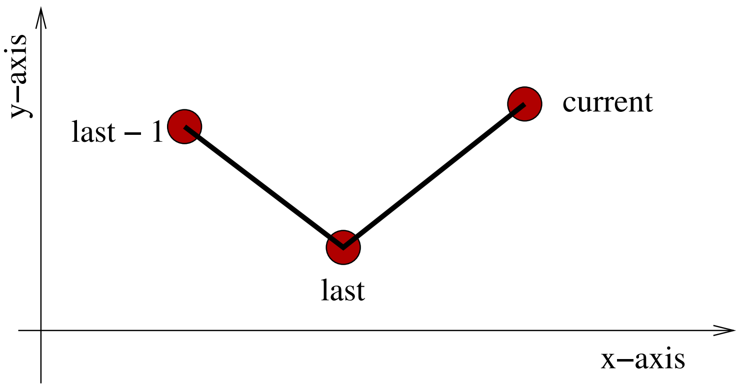

Figure 2: A generic function $f(x)$, where a minimum has been found.

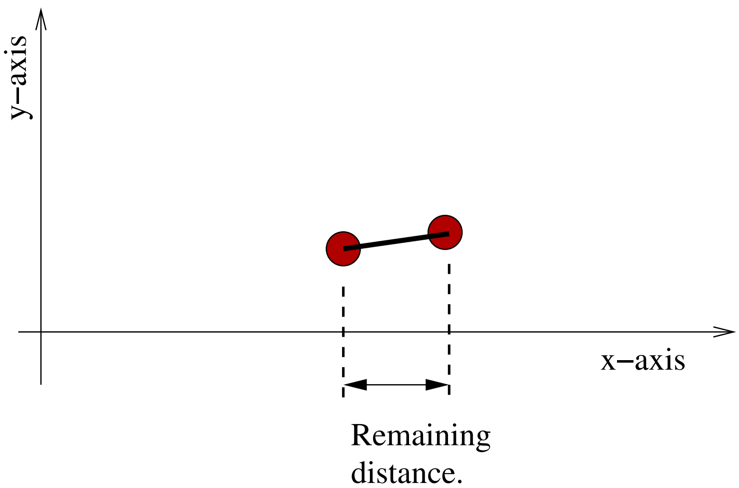

Figure 3: To remaining points that are close together.

Listing 3: Generating a uniform random number

```
import random
random.uniform(-2., 5.)
```

Listing 4: Finding the index of the highest value in a list.

```
lst = [1, 3, 2] # Example list with three values.
i = lst.index(max(lst)) # Returns 1 as expected.
```

Listing 5: Finding the index of the lowest value in a list.

```
lst = [1, 3, 2] # Example list with three values.
i = lst.index(min(lst)) # Returns 0 as expected.
```