ME1 Computing, Consolidation 1: sessions 1-5

Learning outcomes:

- Recap and consolidate topics covered in Sessions 1-5

Please provide feedback at: www.menti.com with code 63 53 57

Before you start

In your H drive create a folder *H:\ME1MCP\Consolidation* and work within it.

Exercise 1: Series expansion

The function *sin(x)* can be evaluated with the Taylor expansion:

$$sin(x) = \sum_{\substack{i=1 \\ i\ odd}}^{N} (-1)^{int(\frac{i}{2})} \frac{x^i}{i!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots$$

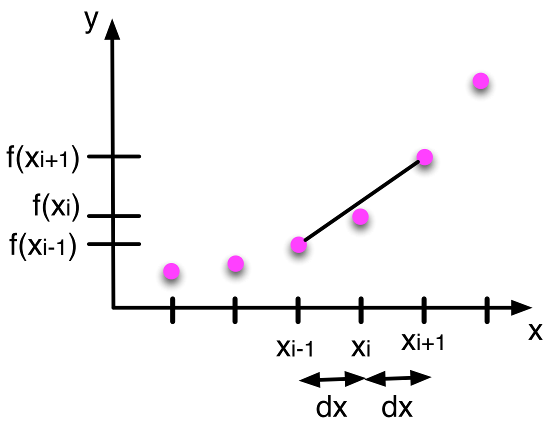
Write a script to compute and plot *sin(x)* vs *x* in the range *x = [0 : 2π]* with step 0.01, for a given *N*.

Exercise 2: Derivative of a function

The file *Bell.txt* contains a set of points representing the Gaussian curve $f(x) = e^{-\frac{x^2}{2}}$ in the range *x = [-4 : 4]* with a step dx = 0.01.
Plot *f(x)* vs *x*.
Compute the derivative of f(x) as:

$$\frac{df(x_i)}{dx} \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2\ dx}$$

for all the points in the given range, apart the first and last points.

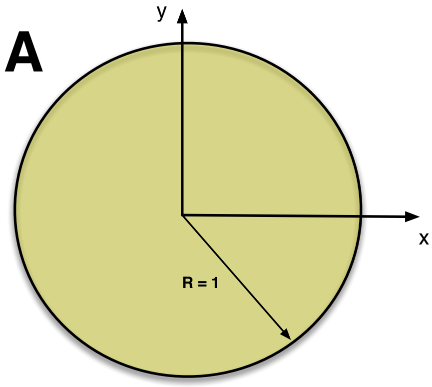


Plot both the computed and the analytical derivative of *f(x)*.

Exercise 3: Ant movement tracing

Write a script to simulate and visualise the random motion of an ant within a closed circular domain.

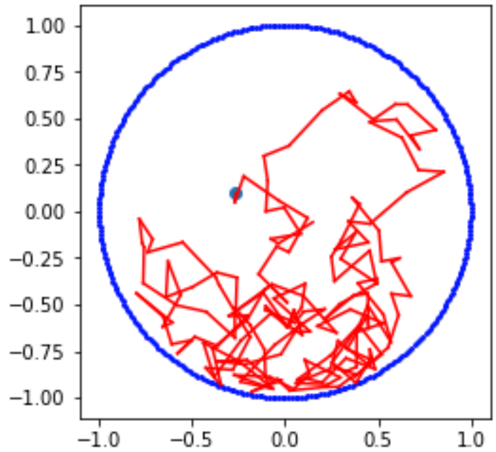
Initially generate **a** random point within the circular domain A and plot it.



Then move the point N times: at every step the ant must jump to a new position by a distance (*dx,dy*). The sizes *dx*, *dy* of the move must be independent random values between -0.2 and 0.2.

If the ant, after moving, leaves the domain, it must bounce back to the current position.

Plot the trace of the ant's moves.



Exercise 4: Find the absolute minimum of a function

The file *Min1.txt* contains a set of points representing the curve:

$$f(x) = 2 \cdot (x - 0.5)^2 - 5$$

in the range *x = [-4 : 4]* with a step dx = 0.01.

Plot *f(x)* vs *x*.

Find the position x_{min} of the minimum value in the given range.

You can verify the position of the minimum by computing the derivative of *f(x)*, using the definition given in Exercise 2. Plot the computed derivative and observe its behaviour where the minimum occurs.

Exercise 5: Find relative minima of a function

The file *Min2.txt* contains a set of points representing a curve *f(x)* in the range *x = [-4 : 4]* with a step dx = 0.01. Plot *f(x)* vs *x*.

Find the positions of all relative minima in the given range.

You can verify the position of the minima by computing the derivative of *f(x)*, using the definition given in Exercise 2. Plot the computed derivative and observe its behaviour where the minima (and maxima too) occurs.

Exercise 6: Dammit I'm mad!

Write a script to establish if a sequence of characters is palindrome.

A word or sentence (excluding any punctuation, upper/lower case and blank spaces) is palindrome if it is the same when read from left to right or right to left.

Input the word to be examined from the keyboard.

d	**a**	**m**	**m**	**i**	**t**	**i**	**m**	**m**	**a**	**d**

Once completed, you can test your script with these examples:

I did, did I?

Don't nod.

I, man, am regal; a German am I?

No mists or frost, Simon.

Did Hannah say as Hannah did?

Exercise 7: Lists and Tuples

The budget of a company (in thousands pounds) for year 2018 is stored in the file *Budget.txt* with the following numerical format (column 1 of the table):

1	*this month*
4	*num. of expenses this month*
5.5	*expenses 1*
15.5	*expenses 2*
2.3	*expenses 3*
4.5	*expenses 4*
3	*num. of incomes this month*
2.6	*income 1*
4.8	*income 2*
1.0	*income 3*
2	*this month*
3	*num. of expenses this month*
2.5	*expenses 1*
	etc.

Write a script to read the file and organise the data in a list of tuples:

(month, total expenses, total incomes)

For every month compute the total savings and store them in the file *Savings.txt*.

Exercise 8: Fractals

Fractals are complex patterns that repeat themselves at different scales. Given an initial point in space, (x_0, y_0), every other point being generated, (x_i, y_i), depends on the previous one (x_{i-1}, y_{-1}).

Write a script to plot *N* points with coordinates (x_i, y_i), generated from the following iterative sequence (also known as Julia set):

$$x_i = y_{i-1} - k * \sqrt{|b * x_{i-1} - c|} \qquad k = \begin{cases} -1 & \text{if } x_{i-1} < 0 \\ 0 & \text{if } x_{i-1} = 0 \\ +1 & \text{if } x_{i-1} > 0 \end{cases}$$
$$y_i = a - x_{i-1}$$

You will need the intrinsic function *abs()*.
Plot all the points in blue.

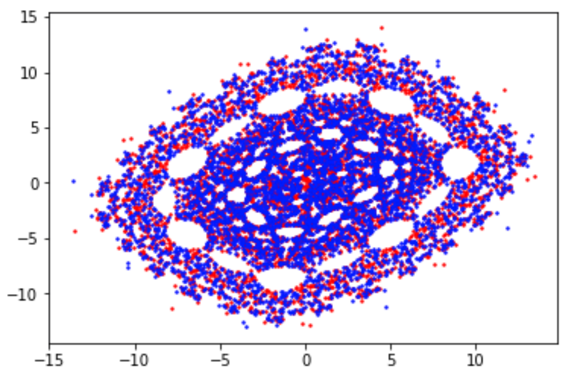
Run the script with:

Setup 1: $a = 0.4$, $b = 1$, $c = 0$, $(x_0, y_0) = (0, 1)$, $N = 20000$
Setup 2/3: $a = 0.8$, $b = 1.3$, $c = 0.9$, $(x_0, y_0) = (0, 2)$, $N = 10000, 50000$

***Bonus* task**

If you want to spice up the graph, plot the odd and even points with different colours.

Enjoy running the script with different parameters. You will notice that for minor changes of *a, b, c* and (x_0, y_0) you get completely different patterns: this is the typical behaviour of a *chaotic system*.



Disclaimer: plotting fascinating fractals, whose formulas you can find on the web, can be addictive.