Assessed Practical 2: Richardson Extrapolation

1)

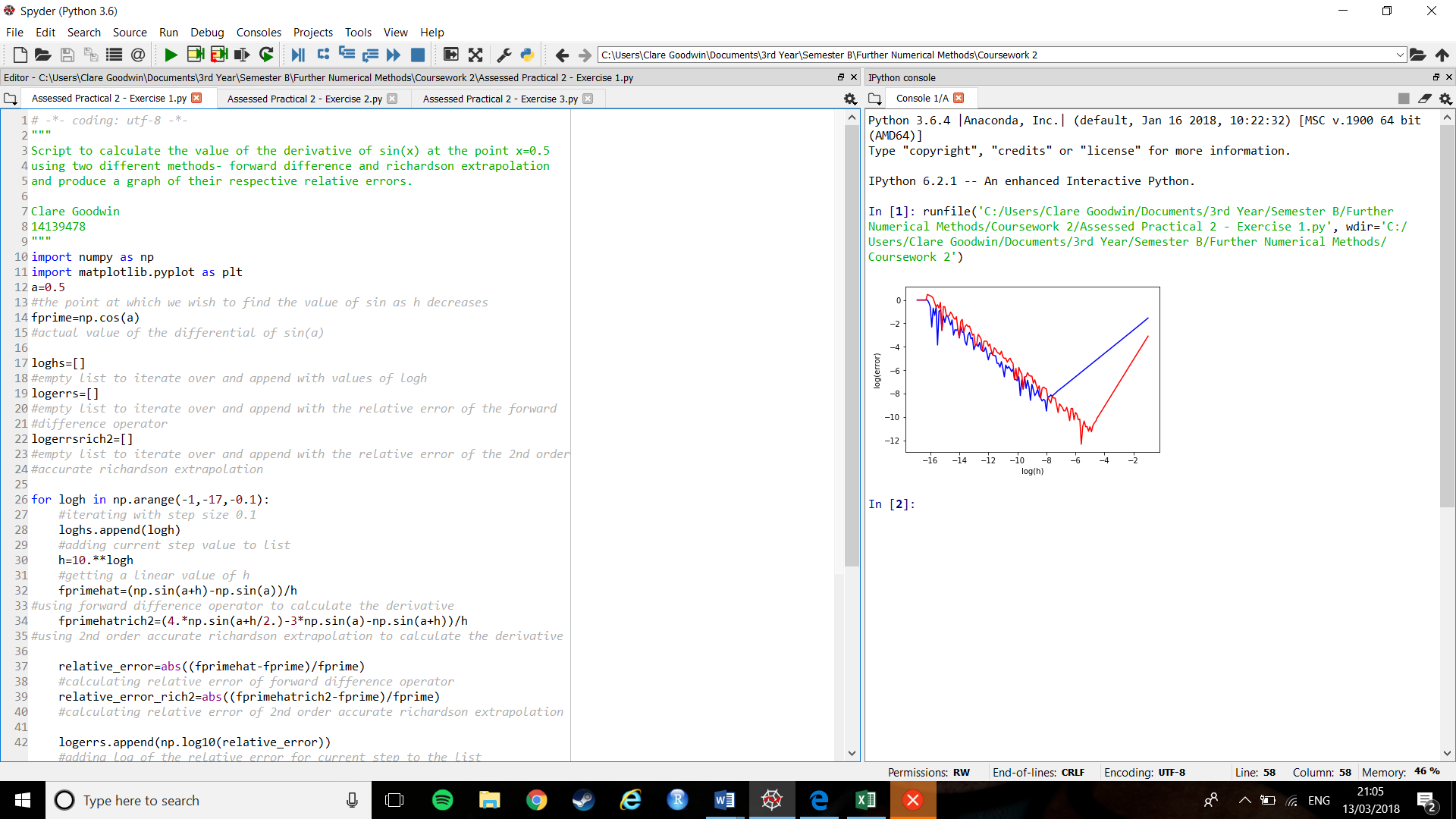


Figure : Graph of relative errors of forward difference operator (blue) against 2nd order Richardson extrapolation (red) for sin(x)

Figure 1 shows the graph obtained using the code found in the file ‘**Assessed Practical 2 - Exercise 1.py**’. The blue line of the graph shows the logarithmic value of the relative error of the values found when calculating the derivative of sin(x) using the forward difference operator against the log of the step size, h; the red line shows the same, but using the 2nd order accurate Richardson extrapolation formula instead. We used the log of the value as a way to effectively ‘zoom in’ on the error term, as we would expect it to be a very small value (which also explains why the values on the graph are negative – the log of very small numbers are negative, and the more negative a value on the graph, the smaller the relative error is). Comparing the two lines in the graph, we notice two things:

Firstly, we can see that, for the first section, the blue line sits lower on the graph, with the red line occasionally, but rarely dropping below it. This implies that the relative error for the forward difference operator is often smaller than that of the 2nd order accurate Richardson extrapolation, and that it begins being the more accurate of the two methods.

Secondly, we notice that the blue line deviates off of its course quite a while before the red line. It is clear that at this deviation the relative error term begins to grow rapidly as its log is becoming less negative. This shows us that the 2nd order accurate Richardson extrapolation is accurate for smaller values of h than the forward difference operator.

2)

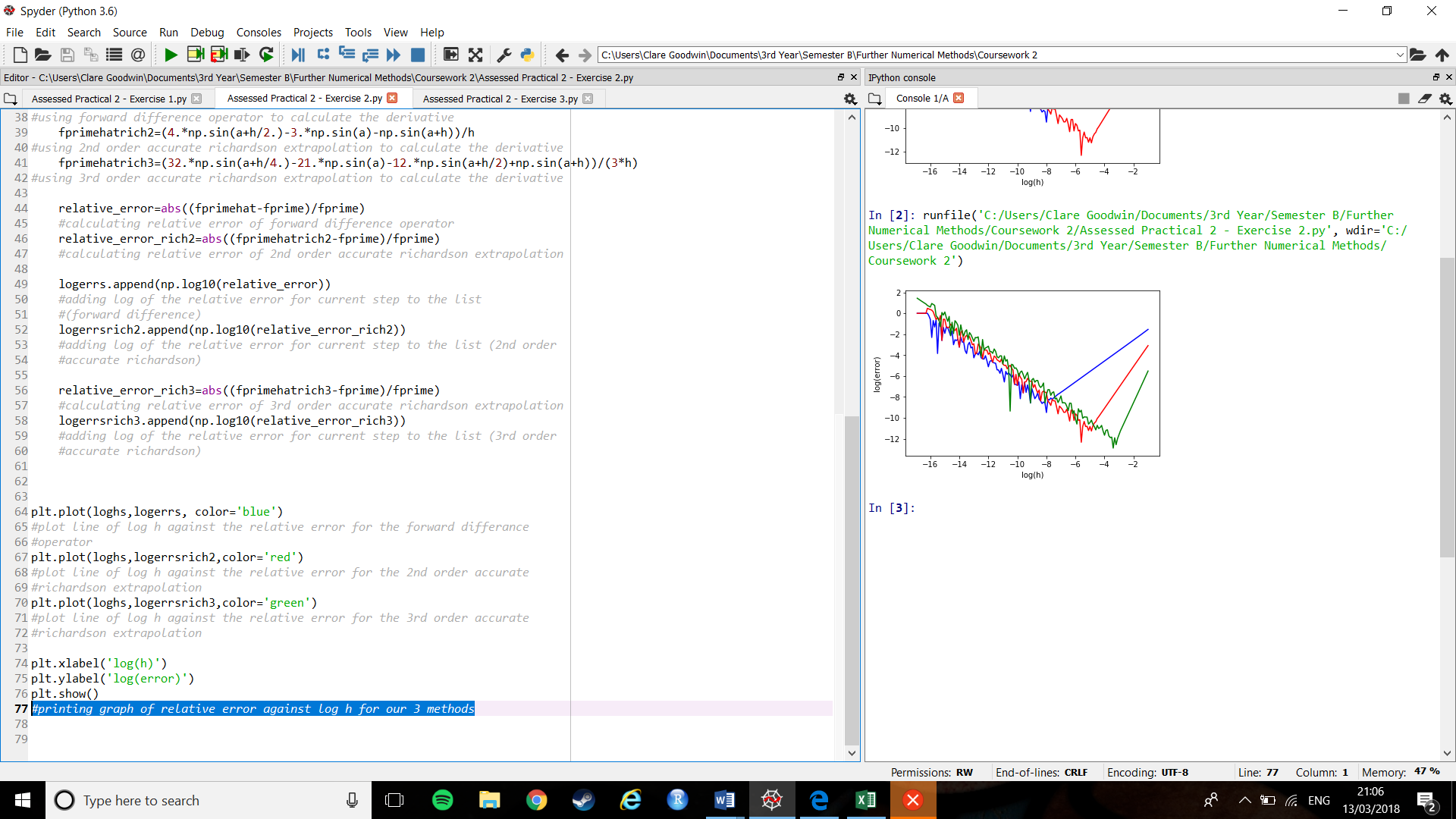


Figure 2: Graph of the relative errors of forward difference operator (blue) against 2nd order (red) and 3rd order (green) Richardson extrapolation for sin(x)

The graph in Figure 2 was obtained using the code found in the file ‘**Assessed Practical 2 - Exercise 2.py**’. The blue line of the graph shows the logarithmic value of the relative error of the values found when calculating the derivative of sin(x) using the forward difference operator against the log of the step size, h; the red line shows the same, but using the 2nd order accurate Richardson extrapolation formula instead; the third shows the same again but for the 3rd order accurate Richardson extrapolation formula.

For this exercise it was required that we derive an equation for a third order accurate Richardson extrapolation. This has been attached separately in the submission in the file ‘**Assessed Practical 2 – Derivation**’.

3)

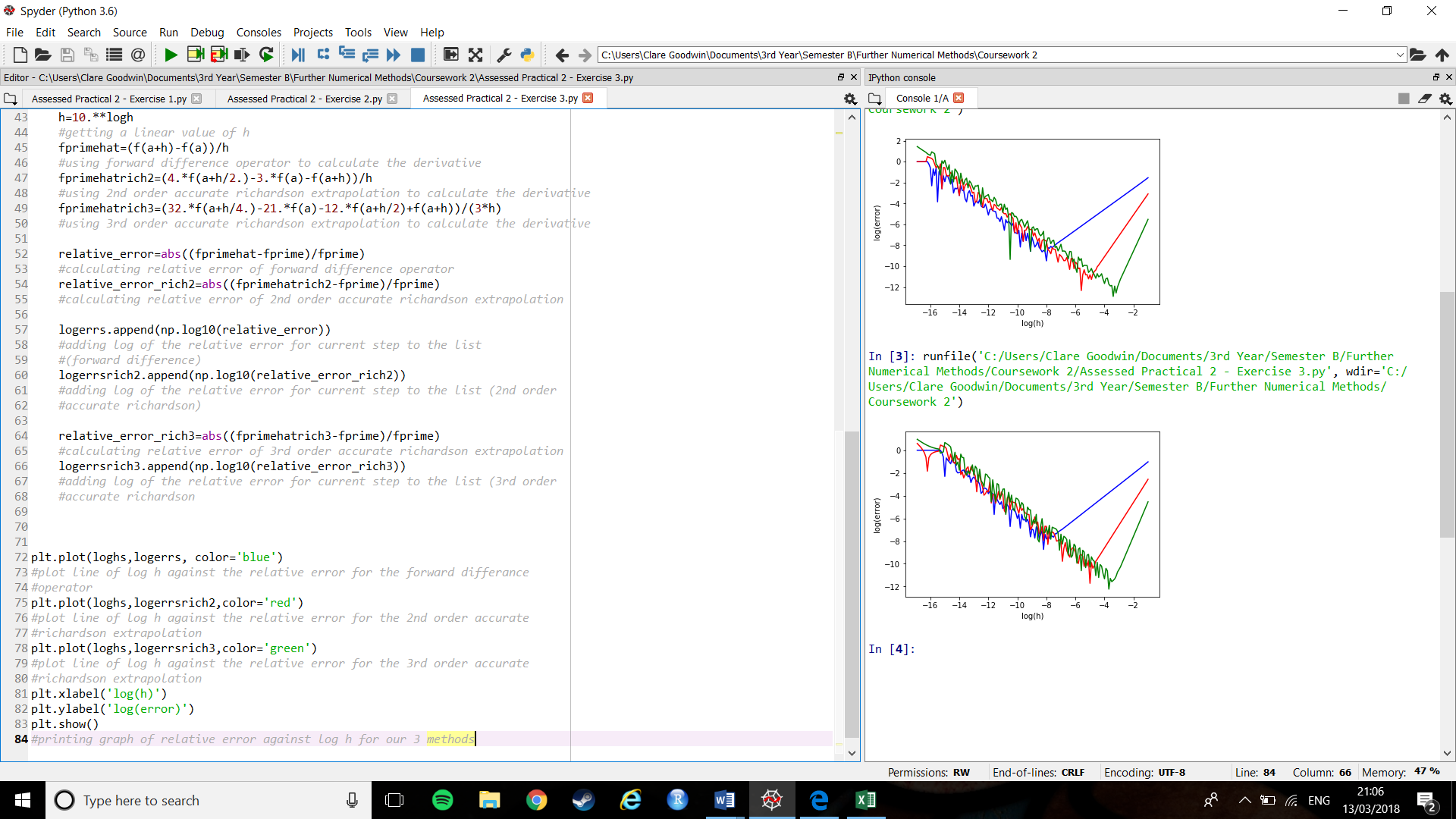


Figure 3: Graph of the relative errors of forward difference operator (blue) against 2nd order (red) and 3rd order (green) Richardson extrapolation for f(x)

Figure 3 depicts the graph obtained using the code found in the file ‘**Assessed Practical 2 – Exercise 3.py**’. The graph shows the log of the relative error for the estimations of the derivative of the function attained using the forward difference operator (blue line), the 2nd order accurate Richardson extrapolation formula (red line), and the 3rd order accurate Richardson extrapolation formula (green line) against the log of the step size, h.  
Looking at Figure 3 in comparison with Figure 2, which was achieved using a much ‘nicer’ function – that of sin(x) – we can see that there are some differences between the two. For example, in Figure 3, for the ‘nastier’ of the two functions, both the red and green lines start above 0, whereas in Figure 2 only the green line does, showing us that the 2nd order Richardson extrapolation gives a larger relative error initially with more difficult functions. It can also be seen that the lines in Figure 3 seems to show more uniform jumps and drops as the step size decreases. However, overall we notice that each of the coloured lines veer off at the same point (where the relative error grows) in both figures, and reaches similarly low values for log(error). This implies that all three equations work similarly well for both difficult functions and relatively simple ones.

Looking at the gradients of the lines we can also see that each of the lines seem to have the same gradient overall, showing that each method achieves smaller relative errors at the similar speeds.