Computational Methods and Modelling

Antonio Attili & Edward McCarthy

antonio.attili@ed.ac.uk
ed.mccarthy@ed.ac.uk

School of Engineering
University of Edinburgh
United Kingdom

Tutorial 9
Numerical Integration

THE UNIVERSITY of EDINBURGH

Numerical integration

Exercise 1

- Implement the Midpoint/Rectangular Rule and execute it to integrate the function $x^2 + 4x$–12 in the domain $-10 < x < 10$.
- Start from the code provided below and also available in Learn. The Midpoint/Rectangular Rule is already implemented in the code below; you have to implement the code to call this function to integrate $x^2 + 4x$–12 in the domain $-10 < x < 10$.

''

```
def calculate_dx (a, b, n):
        return (b-a)/float(n)

def rect_rule (f, a, b, n):
        total = 0.0
        dx = calculate_dx(a, b, n)
        for k in range (0, n):
                total = total + f((a + (k*dx)))
        return dx*total
```

Numerical integration

Exercise 2

- Implement the Trapezoid Rule and execute it to integrate the function $x^2 + 4x – 12$ in the domain $-10 < x < 10$.

Exercise 3

- Implement the Simpson's One Third Rule and execute it to integrate the function $x^2 + 4x – 12$ in the domain $-10 < x < 10$.

Exercise 4

- Implement the 3/8 Simpson's rule and execute it to integrate the function $x^2 + 4x – 12$ in the domain $-10 < x < 10$.

Comparison of techniques

Exercise 5

- Compare the analytical integral value for the equation with the four values obtained by the four techniques above.
- Calculate the true relative error for each technique for the same number of integration intervals and comment on the result.
- Plot the evolution of the integral value as a function of the number of integration intervals for each technique (Hint: you will have to modify each code to run for different values of N and plot the integral obtained for each run. Use arrays to store values and matplotlib to plot Integral v. N).