Assignment 4

Ayden S. McCann

March 2021

Code Listing at the end

Introduction

This assignment explored various methods of numerical integration, both implemented personally and by well-known functions such as scipy.integrate.quad. In order to better understand the overall accuracy and how these methods converge with different Δt values analysis is performed. In Question 1 the solution to a fairly simple integral is explored and in Question 2 the value of Fresnel Integrals, which characterise the intensity of light diffracted through a slit, are explored. Question 3 focuses on the scipy.linalg.eig function and the behaviour of a Spring-Mass system are explored through the analysis of the systems generalised eigenvalue equation and subsequent eigenvectors and eigenvalues.

1 Question 1

The composite versions of the Trapezoidal rule, the midpoint rule, Simpson's rule and the Gaussian quadrature rule were implemented in python and their accuracy tested with the respect to the analytical solution of:

$$\int_0^1 (3 + 2Cos(3\sqrt{x}))dx \tag{1}$$

Solution to integral calculated on Mathematica:

$$3 + \frac{4}{9}(Cos[3] + 3 * Sin[3] - 1) = 2.3037189011462917 \tag{2}$$

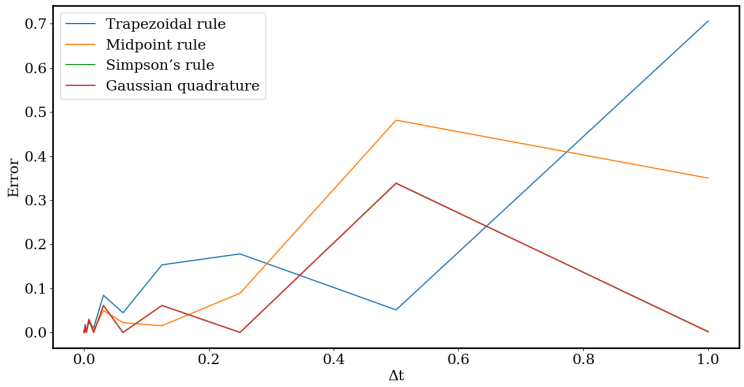

Figure 1: The errors of the 4 different methods for changing Δt values. Note: Simpson's rule values are very similar / identical to Gaussian quadrature for this plot and as a result are hidden behind its line.

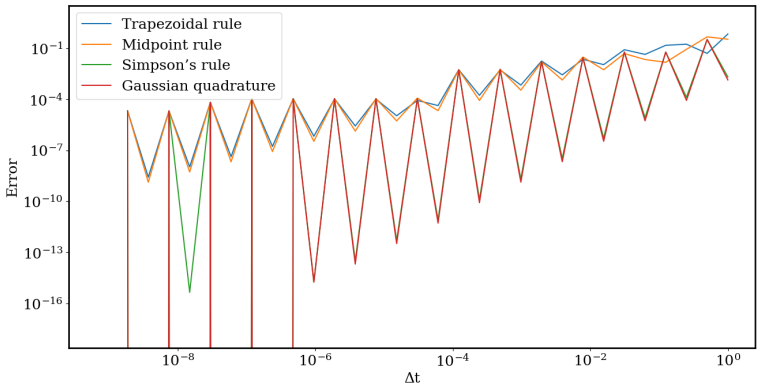

Figure 2: The errors of the 4 different methods for changing Δt values. Displayed on a Log plot to highlight small Δt behaviour.

2 Question 2

According to the documentation at (https://docs.scipy.org/doc/scipy/reference/generated/scipy.integ the scipy.integrate.quad function uses techniques from the FORTRAN77 library QUADPACK.

I cannot explicitly find any further information regarding **which** QUADPACK technique it it uses as there are numerous for different scenarios. All of the methods implemented in the QUADPACK library utilise the Gaussian quadrature method. By my brief investigation i believe that

scipy.integrate.quad should be utilising the QNG method which is a simple non-adaptive routine from QUADPACK.

In order to check whether the results from scipy.integrate.quad agree with those from sc.fresnel the errors were calculated as follows:

Error defined as absolute of (integrate.quad - sc.fresnel)
This was then calculated for 10,000 steps in the range $-5 \leq x \leq 5$. The results are shown below in Figure 3.

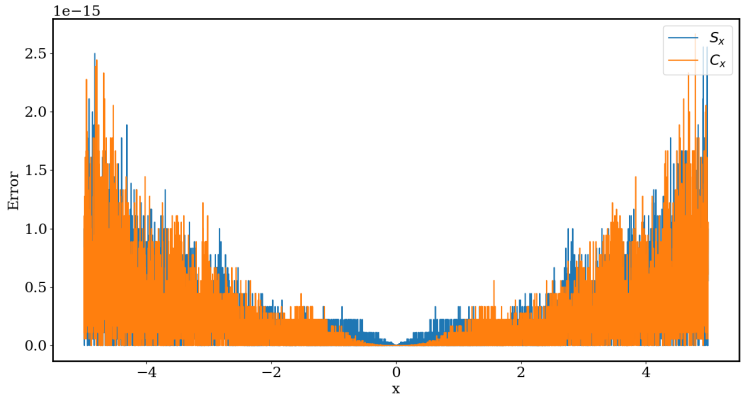

Figure 3: Error between integrate.quad and sc.fresnel values as defined above.

Simply plotting the two functions $C(x) and S(x)$ produce Figure 4:

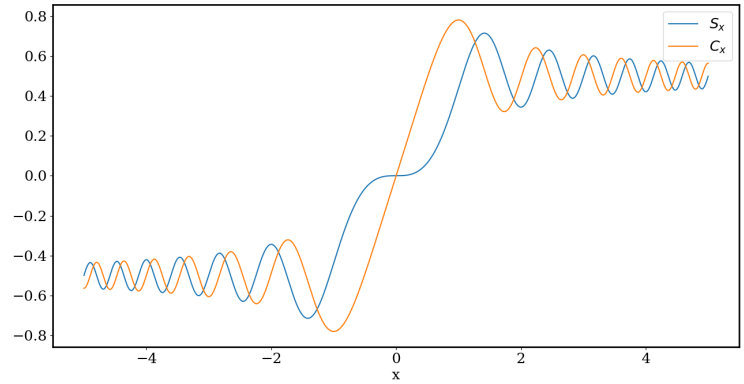

Figure 4: S_x and C_x plotted over -5 $\leq$ x $\leq$ 5.

By plotting the output of these functions parametrically, Figure 5 is produced. This is achieved by plotting the function as a line from $-5 \leq x \leq$

5 but each x value is defined by a pair of coordinates which in this case are $(C(x), S(x))$

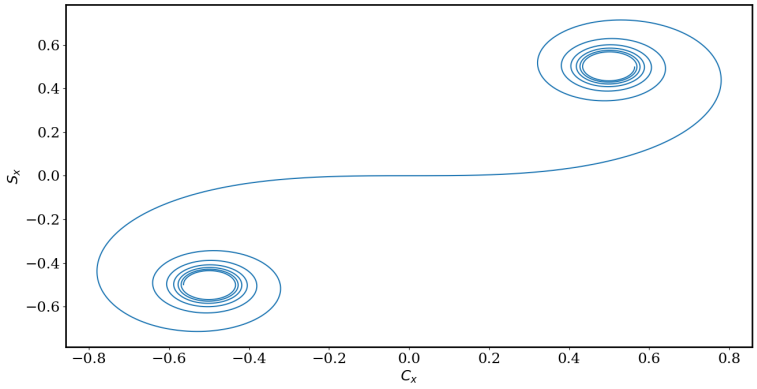

Figure 5: C_x and S_x plotted parametrically on the x and y axes respectively.

3 Question 3

The purpose of question 3 was to investigate the behaviour of a system of masses joined by springs. In order to characterise this system we had to create a mass matrix (**M**), and a stiffness matrix (**K**).

This yields the generalised eigenvalue equation:

$$\mathbf{K} * y = \lambda * \mathbf{M} * y \tag{3}$$

with eigenvalue $\lambda = \omega^2$

scipy.linalg.eig was implemented to solve for the eigenvalues and eigenvectors of this system in 3 configurations:

Case 1. Uniform spring constants, fixed at both ends
Case 2. Uniform spring constants, fixed at only one end
Case 3. A linear spring stiffness profile

By inputting these solved values into the equation of motion (3) the position of a given mass can be plotted with respect to time.

$$x_i(t) = y_i \exp(i\omega t) \tag{4}$$

Below are examples of outputs from this system. They are the plots of the motions of the 1st mass in each eigenvector for the 5 lowest modes over a time interval of 0 to 50.

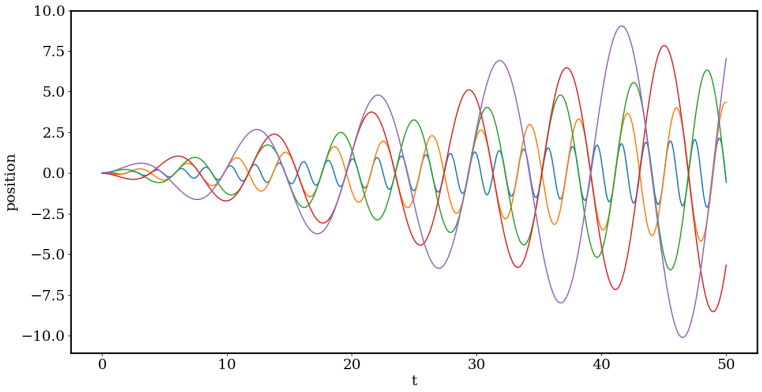

Figure 6: Case 1: The motion of the first mass in its 5 lowest modes.

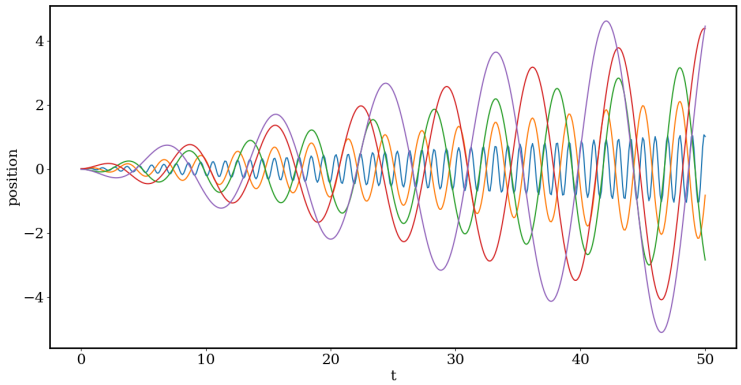

Figure 7: Case 2: The motion of the first mass in its 5 lowest modes.

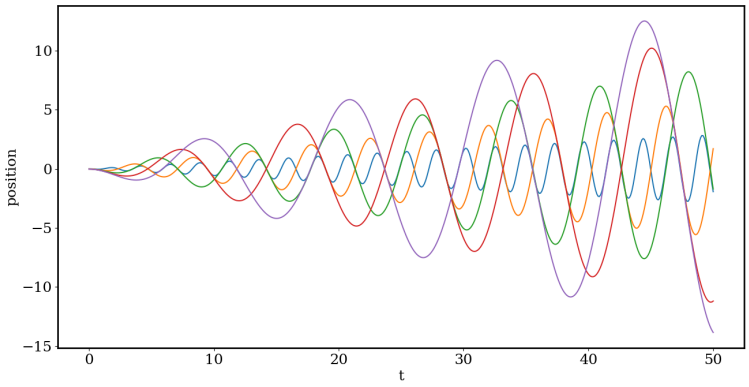

Figure 8: Case 3: The motion of the first mass in its 5 lowest modes.

4 Discussion

As has been demonstrated there are various methods of implementing numerical integration. Research into the QUADPACK FORTRAN77 Library has revealed to be that there are a vast number of differing techniques applicable in differing situations, whether this be for precision or for computational efficiency. The methods coded and analysed by myself in Python are accurate but useful only as an exercise as the many functions existing for python and other coding languages far out-perform them.

There was a fair amount of difficulty incurred in Question 3 when deciding exactly how would be best to analyse the data resulting from obtaining the eigenvalues and eigenvectors. I was presented with the option of plotting either the ith mode, the ith mass and either or both at a given time. If i had more time i would have liked to potentially have plotted all 100 masses offset by a certain amount for a given mode and plotted their trajectories, (perhaps an animation even?) but I feel like the trajectories of a given mass shown in the report are indicative of some of the modes behaviours and my code is capable of being changed to explore any given mass.

5 Code Listing

Question 1 - 1.py

Question 2
2.1py - Checking output agrees with sp.fresnel
2.2.py - Plots

Question 3 - 3.py