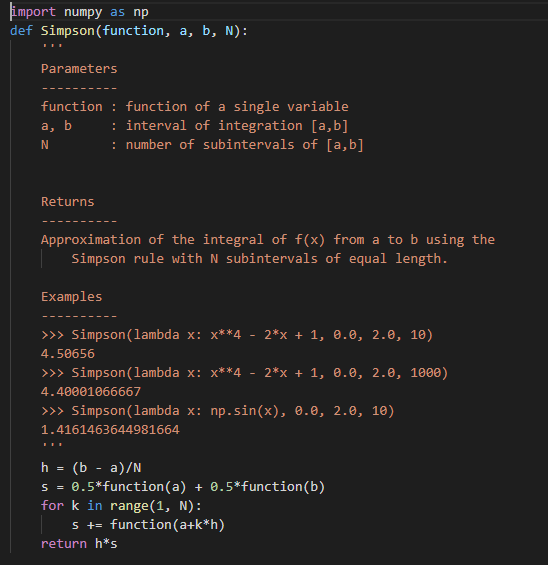
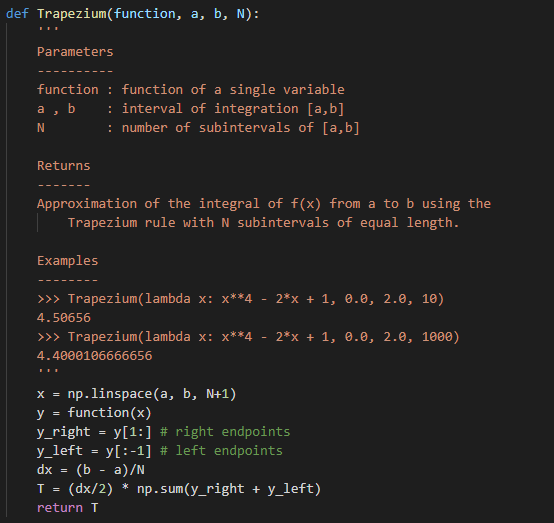
**EXERCISE 2.1: INTEGRATION USING SIMPSON’S RULE**

1. Starting with Example 2.3 in the notes, add a function to perform Simpson’s rule integration.

Code :

1. Use the integral , for which the exact value is 4.4, to compare the accuracy of the trapezium rule and Simpsons rule for the same values of N.

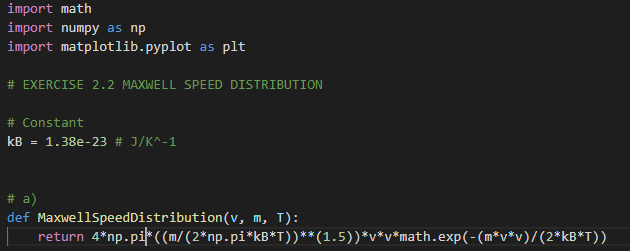
Code implementation for Trapezium’s rule:



Conclusion : when increasing N(number of subintervals of [a,b]), in the case of quadratic functions, the Simpsons method gave the best approximation and the Trapezoidal provided the worst.

**EXERCISE 2.2 MAXWELL SPEED DISTRIBUTION**

1. Write a function that returns a value of f(v) from Maxwell Speed distribution, given an input of v, m and T

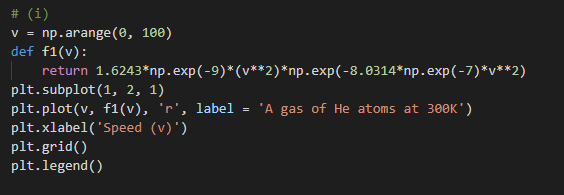
Code :

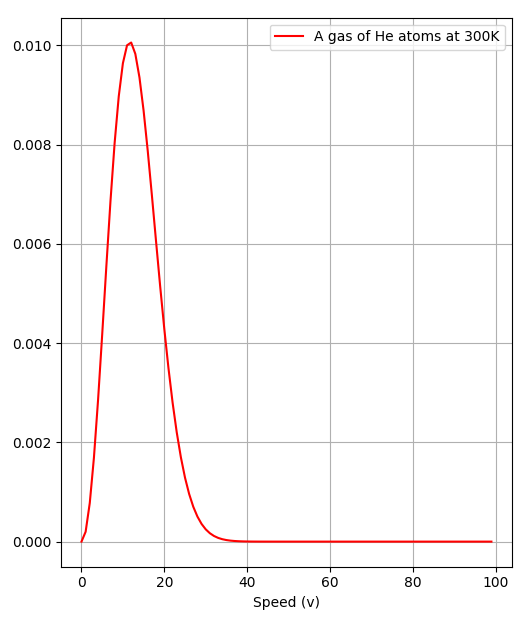
1. Calculate and plot f(v) at the temperatures of 300K an 1000K for:
2. A gas of He atoms, m = 6.65 x kg

Use above function, we have :

f(v) = 1.6243 x v2 exp(-8.0314 x v2)

Code :

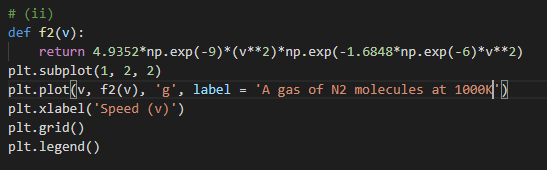


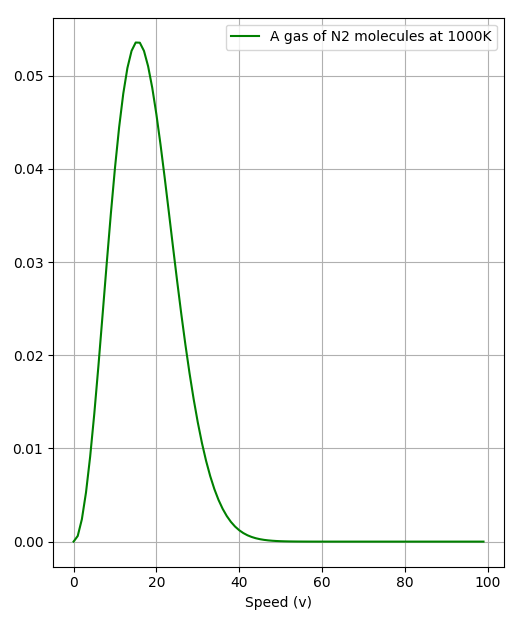
 Graph :

1. A gas of N2 molecules, m = 4.65 x kg

f(v) = 4.9352 x v2 exp(-1.6848 xv2)

Code :



Graph :

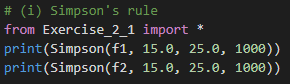
1. Calculate the probability of molecule having a speed between v1 and v2 using
2. Simpson’s rule ( Exercis 2.1)

A gas of He atoms, m = 6.65 x kg

* f(v) = 1.6243 x v2 exp(-8.0314 x v2)
* P(15 < v < 25) = 0.045484531105908416

A gas of N2 molecules, m = 4.65 x kg

* f(v) = 4.9352 x v2 exp(-1.6848 xv2)
* P(15 < v < 25) = 0.44146966504554

Code :

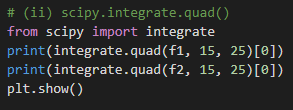
Result :

1. Scipy.integrate.quad ()

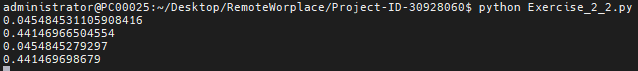
P(15 < v < 25) = 0.04548452792973004

P(15 < v < 25) = 0.44146969867934865

Code :

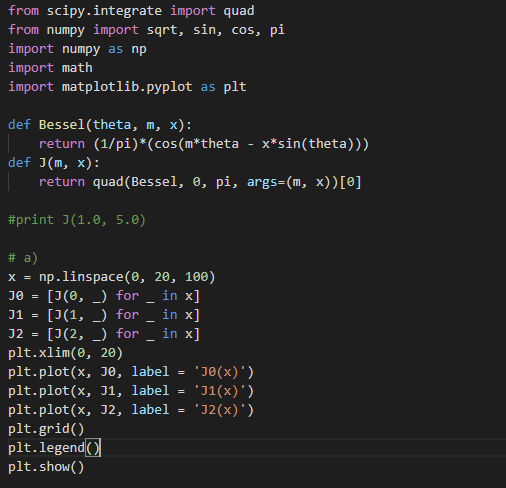


Result :

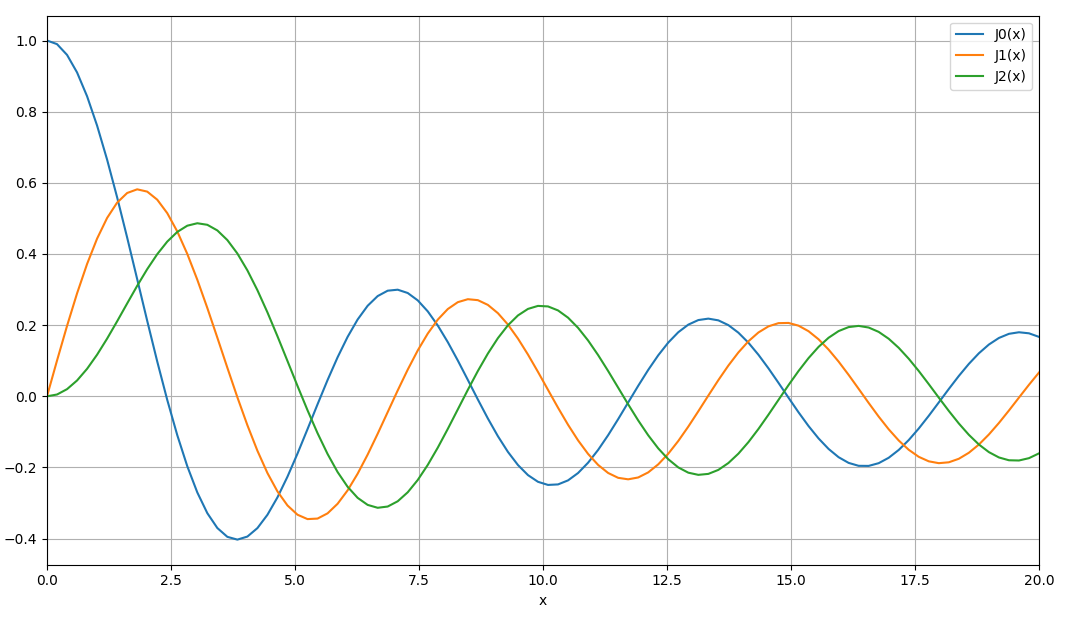


**EXERCISE 2.4: DIFFRACTION LIMIT OF A TELESCOPE**

Write a python function J(m,x) that calculates the value of Jm(x) from Eqn. (5)

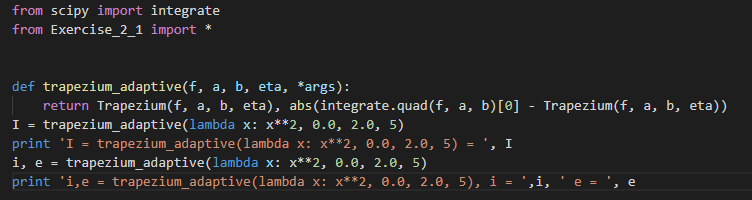
Code :

1. Plot J0(x), J1(x) and J2(x) over the range from x = 0 to x = 20



**EXERCISE 2.5: ERRORS ON INTEGRALS AND ADAPTIVE INTEGRATION**

1. Write a user defined function trapezium\_adaptive(f, a, b, eta, \*args)

Code :

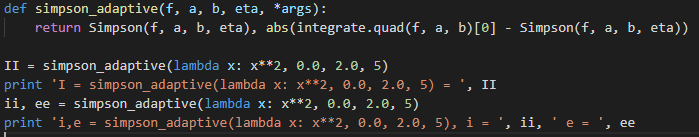
Result :



b)

1. Write a user-defined function simpson\_adaptive(f, a, b, eta, \*args)

Code :

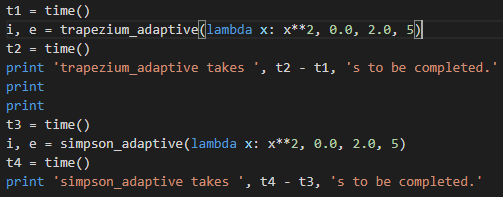


c)

Result :



1. Compare the time it takes for the two functions to calculate a given integral function.

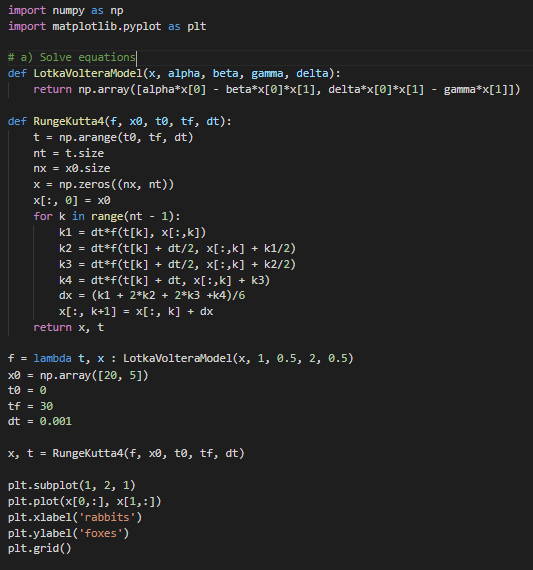
Code :

 Result :

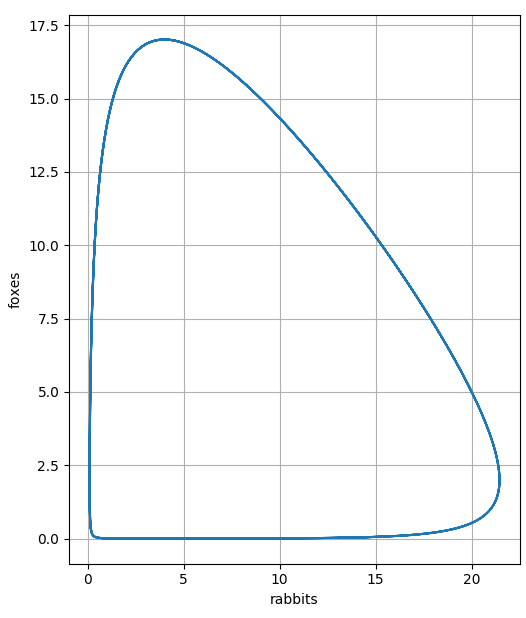
**EXERCISE 3 : THE LOTKA-VOLTERRA EQUATIONS (COUPLED ODES)**

1. Write a program to solve these equations using the fourth-order Runge-Kutta method for the case α = 1, β = γ = 0.5, δ = 2.

Code :



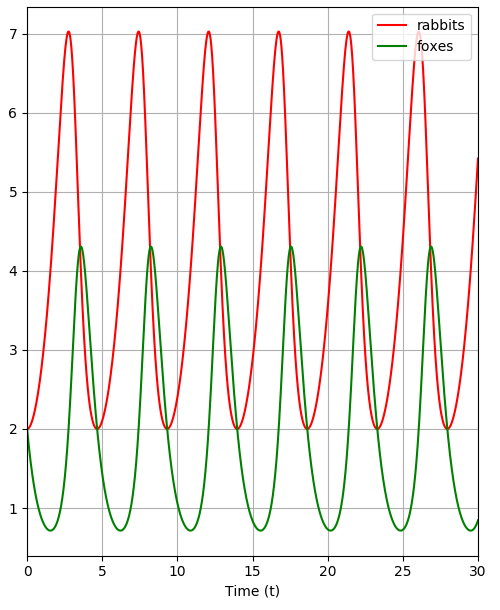
Result :



1. Have the program make a graph showing both x and y as a function of time on the same axes from t = 0 to t = 30, start from the initial condition x = y = 2

Code :



Graph :