B.Sc.(Hons.) Physics
PHC401-32221401
Teacher: Mamta

S.G.T.B. Khalsa College
Mathematical Physics III (2021-22)
Lab Assignment # 2
Integration Module

Due Date and Time : 15.02.2021, 11:59PM
Max. Marks : 40

In this assignment you are required to make a module MyIntegration.py containing functions to compute integration of the type

$$I(f) = \int_a^b f(x)\mathrm{d}x$$

based on the Trapezoidal, Simpson and Gauss-Legendre integration methods.

1. (15 marks) **Theory**
   (a) **Newton Cotes Quadrature :**
      i. Explain the Newton Cotes Quadrature rules. What is the difference between open and closed Newton Cotes? Use method of undetermined coefficients to derive the Trapezoidal, Simpson$_{1/3}$ and Simpson$_{3/8}$ rules for integration.
      ii. Explain how these can be made more accurate by increasing number of intervals and write the composite rules for each of them.
      iii. Discuss the geometrical interpretation of these. What are the conditions on number of intervals for each of them?
      iv. Also discuss the error term in each of these. Can you keep on increasing the number of intervals to reduce the error?

   (b) **Legendre Gauss Quadrature:**
      i. What are Gauss quadrature methods for evaluating integrals? How are Gauss Quadrature methods different from the Newton Cotes Methods?
      ii. Explain how the Gauss quadrature method is closely linked with a set of orthogonal polynomials.
      iii. Explain Legendre Gauss Quadrature methods for evaluation of integral $\int_{-1}^{1} f(x)\,\mathrm{d}\,x$. How will the formula change for the integration $\int_a^b f(x)\,\mathrm{d}\,x$?
      iv. Explicitly derive the 2-point quadrature formula for this method.
      v. Explain n-point composite quadrature formula.

2. (5 marks) **Pseudocode**
   (a) Write the pseudocode to estimate $I(f)$ numerically by I_{num} using Trapezoidal and Simpson$_{1/3}$ methods such that the step size is continuously reduced so as to achieve the integral accurate to d number of significant digits. Make sure that you do not repeat the function evaluation at a point.

   (b) Write the pseudocode
      i. to approximate the integral $I(f) = \int_a^b f(x)\,\mathrm{d}\,x$ by $I_{nm}(f)$ using n-point composite Gauss Legendre quadrature with m number of subintervals.
      ii. to approximate the integral $\int_a^b f(x)\,\mathrm{d}\,x$ using n-point composite Gauss Legendre quadrature with a relative tolerance '*tol*'. The interval [a, b] is iteratively halved into subintervals (i.e. number of subintervals is doubled starting from 1) until relative error between last two iterates is less than *tol* or number of subintervals reach a maximum value. The output should be the value of integral and the number of sub- intervals required to achieve the desired accuracy.

3. (15 marks) **Programming**

   (a) Make a module MyIntegration.py that contains six functions for approximating the integral $I(f)$. This will include two functions for each of the following methods : Trapezoidal, Simpson$_{1/3}$, and Legendre Gauss Quadrature. Your module should include comments including detailed description of each function (explaining the input and output parameters) These functions are described below:

i. **MyTrap and MySimp**:
Python Functions for integrating a given function (either in expression or data form) using the Trapezoidal and Simpson$1_{1/3}$ methods with n number of panels. The input to the function should be the integrand, limits of integration and the number of panels(or intervals) and output should be the value of integral. The number of intervals and the corresponding integrals could be arrays.

ii. **MyTrap_tol and MySimp_tol**:
Python Functions to integrate a given function $f(x)$ using Trapezoidal and Simpson$1_{1/3}$ methods such that the step size is continuously reduced so as to achieve the integral accurate to d number of significant digits (Note that you should take tolerance $= 0.5 \times 10^{-d}$). The input to the function should be the integrand, limits of integration, and the accuracy (either in the form of number of correct significand digits or the tolerance) and a maximum number of intervals. The output should be the value of integral and the number of intervals required to achieve the desired accuracy or the statement that the required tolerance could not be achieved with the maximum number of intervals.

iii. **MyLegQuadrature(**f, a, b, n, m**)**:
Python Function to approximate the integral $I(f)$ using $n-$point composite Gauss Legendre quadrature with m number of subintervals.

iv. **MyLegQuadrature_tol(**f, a, b, n, tol, $m_{\max}$**)**:
to approximate the integral $I(f)$ using n-point composite Gauss Legendre quadrature with a relative tolerance '*tol*'. The interval [a, b] is iteratively halved into subintervals (i.e. number of subintervals is doubled starting from 1) until relative error between last two iterates is less than *tol* or number of subintervals reach a maximum value of $m_{\max}$. The output should be the value of integral and the number of subintervals required to achieve the desired accuracy or the statement that the required tolerance could not be achieved with the maximum number of subintervals.

Note that

1. In case you can make a single function that works for both – fixed number of intervals and fixed tolerance, you are most welcome to do that. In that case your module will have only three functions– one for each method.
2. In both functions for Gauss quadrature, use the function ***numpy.polynomial.legendre.leggauss*** or ***scipy.special.roots_legendre*** for computing the points and weights.

(b) Validate the functions MyTrap, MySimp and MyLegQuadrature by showing that

i. the trapezoidal method with one interval gives exact result for a linear function but not for a polynomial of order 2.

ii. the Simpson method with two intervals gives exact result for a polynomial of degree less than or equal to 3 but not for a polynomial of order 4.

iii. the n-point ($n = 2$) quadrature with only one sub-interval i.e. $m = 1$ gives exact result for a polynomial of degree less than or equal to $2n - 1$ but not for a polynomial of degree $2n$. Verify for $n = 2$ and $n = 4$.

For this verification, evaluate $\int_1^2 f(x)dx$ with $f(x) = 1$, $f(x) = x$, $f(x) = x^2$, etc. You should input these functions from terminal and not define in your module.

(c) Write a separate program Integration-2020PHYxxxx.py (xxxx being the last four digits of your roll number. Physical Science students will use 2020PSC instead of 2020PHY) that imports the module MyIntegration and use the functions MyTrap and MySimp in your module to perform the following tasks:

i. Estimate the value of π by approximating the integral

$$I = \int_0^1 \frac{1}{1 + x^2} dx = \frac{\pi}{4} \tag{1}$$

with I_n i.e. using n intervals. Use $n = 2m$, $m = \{1, 2, 3, \ldots, 16\}$ for trapezoidal and Simpson$_{1/3}$ and store the corresponding values of π obtained as an array my_pi(n).

ii. Plot my_pi(n) as a function of n for both trapezoidal and Simpson$_{1/3}$. Also plot a straight line corresponding to value of π evaluated by *math.pi* on the same plot.

iii. Extend your program to evaluate the error $e(n) = |(\text{my\_pi}(n) - \pi)|$ (where π is evaluated as before) for each n and plot $e(n)$ as a function of n. Use appropriate range and scale in each graph and label it properly. If $h = (b - a)/n$ is the corresponding interval size, plot $\ln(e)$ as a function of $\ln(h)$ and interpret the result.

(d) Use the functions MyTrap_tol and MySimp_tol to estimate the value of π correct to d number of significant digits. Print the output in a tabulated format as shown in Table 1.

Table 1: *Values of π computed numerically accurate to __ significant digits alongwith number of intervals n.*

method	my_pi(n)	n	$E = \|\text{my\_pi}(n) - \pi\|/\pi$
Trapezoidal			
Simpson$_{1/3}$			

(e) **Legendre Gauss Quadrature:**
Extend your program to use the function MyLegQuadrature for performing following tasks:

i. Estimate the value of π by approximating the integral (1) with $n-$point quadrature formula with $n = 2, 4, 8, 16, 32, 64$ and for each of them use $m = 1, 2, 4, 8, 16, 32$. Store these values in a matrix pi_quad(n, m) and print in a tabulated format as shown in Table 2. Also store this

Table 2: *Values of π approximated by Gauss Legendre Quadrature with $n-$point formula with m subintervals.*

n	$m = 1$	$m = 2$	$m = 4$	$m = 8$	$m = 16$	$m = 32$
2	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$
64	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

data in a text file named pi_quad-xxxxa.dat where xxxx are the last four digits of your class roll number.

ii. Plot the value of pi_quad(n, m) obtained numerically above as a function of n for $m = 1$ and $m = 8$. Also plot pi_quad(n, m) as a function of m for $n = 2$ and $n = 8$. Further, plot a straight line corresponding to value of π evaluated by *math.pi* on the same plot.

iii. Extend your program to evaluate the error $e(n, m) = |(\text{pi\_quad}(n, m) - \pi)|$ for each m, n and plot $e(m, n)$ as a function of n (for $m = 1$ and $m = 8$) and also as a function of m (for $n = 2$ and $n = 8$). Use appropriate range and scale in each graph and label it properly.

(f) Use the function MyLegQuadrature_tol to estimate the value of π by approximating the integral stated in above question with tolerance of 0.5×10^{-d}, with $d = 1, 2, \ldots, 8$. Record the result in a tabulated format as given in Table 3.

Table 3: *Values of π approximated by Gauss Legendre Quadrature with $n-$point formula with tolerance tol.*

	$tol = 5 \times 10^{-2}$		$tol = 5 \times 10^{-3}$		$tol = 5 \times 10^{-4}$		$tol = 5 \times 10^{-5}$		$tol = 5 \times 10^{-6}$		$\ldots$
n	π	m	π	m	π	m	π	m	π	m	
2	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
32	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	

(g) Use inbuilt functions *fixed_quad* and *scipy.integrate.quadrature* for Gauss Quadrature for above task and compare your results.

4. (5 marks) **Discussion**
   Interpret and discuss your results and graphs and compare the three methods along with the order of error in each case.