PHYS2160 Introductory Computational Physics
2021/22 Exercise 4

1. A reasonable algorithm to calculate the function $\sin x$ is to use the approximation

$$\sin x \approx S(x;n) = \sum_{i=1}^{n} \frac{(-1)^{i-1} x^{2i-1}}{(2i-1)!}$$

Write a Python program to investigate the error in computing $\sin x$ as the finite sum $S(x, N)$ by doing the followings:

(a) implement the function `sinsum1` that returns the finite sum $S(x;n)$ and the number of terms n in the sum for a given value of x so that the last term summed is less than 10^{-7} of the sum, i. e.

$$\left| \frac{(-1)^{n-1} x^{2n-1}}{(2n-1)!} \right| \leq 10^{-7} \left| \sum_{i=1}^{n} \frac{(-1)^{n-1} x^{2n-1}}{(2n-1)!} \right|,$$

(b) use the function `sinsum1` to generate a table of values of $S(x, n)$, n, and ϵ_r for $x = 0.1, 0.2, \ldots, 1.0$ where ϵ_r is the relative error given by

$$\epsilon_r = \left| \frac{S(x,n) - \sin x}{\sin x} \right|$$

with the value of $\sin x$ obtained from the built-in function `math.sin(x)`,

(c) implement the function `sinsum2` that returns the finite sum $S(x;n)$ for given values of x and n,

(d) use the function `sinsum2` to calculate the finite sum $S(x, n)$ and its absolute error

$$\epsilon_a = |S(x,n) - \sin x|$$

together with the value of $\sin x$ obtained from the built-in function `math.sin(x)` for $x = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50$ as n increases from 1 to 20,

(e) use the `matplotlib.pyplot` function `semilogy` to plot the magnitude of the absolute error $|\epsilon_a|$ versus the number of terms n in the sum with log scaling on the y-axis for different values of x on the same graph using the data obtained in part (d).

2. Write a Python program that evaluates the integral

$$I = \int_0^{\pi} \sin x \, dx$$

using the trapezoidal and Simpson's rule with N slices where N = 4, 8, 16, 32, 64, 128, 256, 512, and 1024. Your program should output a table of the integral I with its error (i. e. the absolute difference from the exact value of the integral) for different values of N.

3. Write a Python program that calculates the value of the integral

$$I = \int_0^1 e^{-x}\, dx$$

using the trapezoidal and Simpson's rules with N slices where $N = i^2$ for $i = 2, 4, 6, \ldots, 80$. Your program should compute the relative error $\epsilon = |(I_e - I)/I_e|$ of the integral I in each case where I_e is the exact value of the integral and then make a log-log plot of the relative error ϵ versus the number of slices N in each case.

4. Write a Python program that uses adaptive Simpson's rule to compute the value of the integral

$$I = \int_0^1 \sin^2(\sqrt{100x})\, dx$$

to an accuracy of 10^{-8} with an initial number of slices of 10. Your program should output the value of the integral and the number of slices used to obtain the result.

5. Write a Python program that uses the Romberg integration to compute the the value of the integral

$$I = \int_0^1 \sin^2(\sqrt{100x})\, dx$$

to an accuracy of 10^{-8} with an initial number of slices of 10. Your program should output a triangular table of values of all the Romberg estimates of the integral in the same format as in Fig. 6.5 of the lecture notes.

6. Planck's law of thermal radiation tells us that the total energy per unit area radiated by a blackbody per unit time at temperature T is

$$I = \frac{2\pi h}{c^2}\int_0^\infty \frac{f^3}{(e^{hf/k_BT}-1)}df = \frac{2\pi k_B^4 T^4}{c^2 h^3}\int_0^\infty \frac{x^3}{(e^x - 1)}dx$$

where h is the Planck constant, c is the speed of light in vacuum, and k_B is the Boltzmann constant. On the other hand, experiment revealed that the total intensity I of the radiation emitted by a blackbody follows the Stefan-Boltzmann law

$$I = \sigma T^4$$

where σ is the Stefan-Boltzmann constant. Write a Python program that evaluates the integral in the first expression using Simpson's rule with 100 slices and then uses this integral to compute the value of the Stefan-Boltzmann constant (in SI units) to five significant figures. Your program should output the resultant value of the Stefan-Boltzmann constant with its theoretical value.

(Hint: To evaluate the improper integral $\int_0^\infty f(x)\, dx$, we can use the change of variable $z = x/(1+x)$ to obtain

$$\int_0^\infty f(x)\, dx = \int_0^1 \frac{1}{(1-z)^2} f\left(\frac{z}{1-z}\right) dz$$

which can be evaluated by any techniques of numerical integration.)

7. Write a Python program that computes the first-order derivative $f'(x)$ of the function $f(x) = \cos x$ at 9 uniformly spaced points for $0 \leq x \leq \pi$ using the forward difference, backward difference, and central difference based only on the values of $f(x)$ at 11 uniformly spaced points for $-\pi/8 \leq x \leq 9\pi/8$. Your program should output a table of each derivative with its error (i. e. the absolute difference from the exact value of the derivative) obtained by these methods for different values of x.

8. Write a Python program that computes the second-order derivative $f''(x)$ of the function $f(x) = \cos x \sinh x$ at 51 uniformly spaced points for $0 \leq x \leq \pi/2$ using the central difference based only on the values of $f(x)$ at 53 uniformly spaced points for $-\pi/100 \leq x \leq 51\pi/100$. Your program should output a table of the derivative with its error (i. e. the absolute difference from the exact value of the derivative) for different values of x.

9. Below table displays the values of the function $f(x) = \exp(\sin x)$ at several points:

x	0	0.36	1.22	2.15	2.78	3.62	4.27	5
y	1	1.42229	2.55768	2.30919	1.42442	0.63105	0.40505	0.38331

Write a Python program that performs linear interpolation on the above data to find the values of $f(x)$ at $x = 0.1, 0.2, 0.3, \ldots, 4.9$. Your program should plot the interpolated values and exact values of $f(x)$ as well as the data for interpolation versus x for $0 \leq x \leq 5$ on the same graph.