AA 2021/2022 Computational Methods
Lesson 4 - Numerical Integration

Almaz Khabibrakhmanov*, Mario Galante†, Alexandre Tkatchenko

Theoretical Chemical Physics group, FSTM,
Campus Limpertsberg, University of Luxembourg

Exercises

Part 1 - Midpoint and Trapezoidal Rules

1. Download the file `midpoint.py` from Moodle.

2. Take $f(x) = sin(x)$ as an example function and calculate its integral on $[0, \pi]$ using `Midpoint` function for $n = 5$. Calculate the error of integration as:

$$\Delta = |S_{Mid} - S_{an}|, \tag{1}$$

   where S_{Mid} is the value obtained from `Midpoint`, and $S_{an} = \int_0^\pi sin(x)\, dx$ is the analytical value of the integral.

3. Create a `numpy` 1D array, `N`, containing the following values for the number of sub-intervals, n, that partition $[a, b]$: 5, 10, 20, 30, 50, 100.

4. Use a `for` loop to calculate again the integral of $\sin(x)$ for $x \in [0, \pi]$ for each value of n in the array `N`. At every iteration append your calculated value of the integral S_{Mid} to the array `SmidSin` and the error Δ to the array `EmidSin`. Print n, S_{Mid}, and Δ at every iteration. Look at the results. What do you see?

5. Following the example for the midpoint rule therein, define a function `Trapezoid` that implements the trapezoidal rule for numerical integration:

$$\int_a^b f(x)\, dx = \frac{f(a) + f(b)}{2} \cdot h + \sum_{i=1}^{n-1} f(x_i) \cdot h + O(h^2) \tag{2}$$

   Keep in mind that here $\{x_i\}_{i=0}^n$, with $x_0 = a$ and $x_n = b$, define n sub-intervals, $[x_i, x_{i+1}]$, that form a uniform partitioning of $[a, b]$.

6. Repeat 4) using `Trapezoid` function for the calculations. In this case, name the arrays for the integral values and errors as `StrSin` and `EtrSin`.

7. Define an array, `h`, containing the size of the integration step, $(b - a)/n$, for each value of n in the array `N`. Plot `h` against both `EmidSin` and `EtrSin` in one plot. Recall your knowledge of plotting from the previous lessons: specify the legend and add axis titles. Can you notice any difference between the midpoint and trapezoidal rules from this plot?

8. Now repeat points 4) - 6) for the function $g(x) = 3x^2$ on $[0, 2]$. Suggested names for your arrays of errors would be `EmidParab` and `EtrParab` in this case for midpoint and trapezoidal rule, respectively. Compare your plot with the one from 6). What is your conclusion now?

*almaz.khabibrakhmanov@uni.lu
†mario.galante@uni.lu

Part 2 - Let's calculate something more interesting

1. Below, you see the *pseudocode* which describes the algorithm to calculate the integral of a given function $f(x)$ on $[a, b]$ interval with a predefined tolerance ε using the midpoint rule. Implement this algorithm as a Python code. Add printing of the integral and the error at every iteration to get more information from the program.

Algorithm 1 `Integrator`

```
define the function f(x)
set the values for a and b
set the value for the tolerance, ε
set the initial number of points in partitioning n = 5
calculate the integral S by calling Midpoint(f,a,b,n)
define the absolute error absErr = abs(S)
define the relative error relErr = absErr/S
set the maximum number of iterations M = 20
set the iteration counter j = 1
while j < M and absErr > ε · S do
    j = j + 1
    Sold = S
    n = 2n
    S = Midpoint(f,a,b,n)
    absErr = abs(S - Sold)
    relErr = absErr/S
end while
return S
```

2. Take $f(x) = \sqrt{1 - x^2}$ and integrate it over $[0, 1]$ interval with a tolerance $\varepsilon = 10^{-6}$ using `Integrator`. Print the result multiplied by 4 and look at it. What does this number remind you of?

3. Explain mathematically why you have obtained that number when integrating $f(x) = \sqrt{1 - x^2}$ over $[0, 1]$ and then multiplying this integral by 4. *Hint:* try plotting $f(x)$ if you have no idea.

4. Try to reduce the tolerance down to $\varepsilon = 10^{-12}$ and execute again. What have you noticed? Have you managed to converge to desired accuracy within 20 iterations? Suggest an explanation.

Part 3 - Simpson's Rule (optional)

1. Implement the Simpson's rule:

$$\int_a^b f(x)\, dx = \frac{h}{3} \cdot \left(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_{2i}) + 4\sum_{i=1}^{n} f(x_{2i-1}) \right) + O(h^4) \tag{3}$$

   Be careful with the grid, keep in mind that the odd number of points should be used for the partition of $[a, b]$. Add to your code an `if` that checks this condition and redefines $n := n + 1$ if n is even.

2. Repeat 4) - 8) from the Part 1 for Simpson's rule. Compare the results with those obtained for midpoint and trapezoidal rules. What do you observe for $f(x) = 3x^2$? Try also to integrate $f(x) = x^3$ and $f(x) = x^4$ over $[0, 2]$. Analyze the results and make the conclusion about the accuracy of Simpson's method for polynomials.