MA305 – Classwork # 6 → (Homework)
Modules in Python: Numerical Integration

In this classwork you will learn how to create your own modules (collection of functions) in Python and use them to approximate the integrals.

$$\text{(a)}\ \int_0^1 (2x+1)(x^2+x+1)^3\,dx = 20 \qquad \text{(b)}\ \int_0^b e^{-x^2}\,dx = \frac{\sqrt{\pi}}{2}\text{erf}(b)$$

---

1. Download the file **cw6.zip** from your course Canvas, save it in some appropriate directory (e.g., `MA305/Classwork/CW6`), unzip it and give a look to each file.

```
$ unzip cw6.zip
$ cat cw6.py
$ head num_int.py
$ tail my_funs.py
```

2. The code `cw6.py` approximates the integral (a) using three methods. Run it using $n = 10, 100$ and 1000. What did you observe. It seems, the `middle_sum()` is not working properly.

3. Check the function `middle_sum(f,a,b,n)` in the file `cw6.py` (lines 33-37) and correct it by filling the missing expressions inside the `for loop` as shown below.

- Midpoint Rule: $\displaystyle\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f(\bar{x}_i)\Delta x, \quad \Delta x = \frac{b-a}{n},\ \bar{x}_i = a + \left(i + \frac{1}{2}\right)\Delta x$

```
1 def middle_sum(f,a,b,n):
2     """Approximates integrals by summing the area of the rectangles with
      height taken from the midpoint."""
3     dx=(b-a)/n
4     msum=0
5     for i in range(n):
6         xibar = a+(i+0.5)*dx
7         msum += f(xibar)
8     return dx*msum
```

Run the corrected code **cw6.py** now. Which method is better?

4. Now copy the file **cw6.py** to **cw6a.py**, delete all the functions `f, left_sum, right_sum, middle_sum` in **cw6a.py** (lines 11-40) and add the following two import statements just after the document string (line 9) of the file.

```
import my_fun
import num_int
```

Make the necessary changes in the calling sequence of the function, `f, left_sum, right_sum, middle_sum` as

```
approx1=num_int.left_sum(my_funs.f,a,b,n)
approx2=num_int.right_sum(my_funs.f,a,b,n)
approx3=num_int.middle_sum(my_funs.f,a,b,n)
```

and run the code **cw6a.py**. You should get the same results as in **3**.

5. Now import the function $g(x) = e^{-x^2}$ from `my_funs` and `left_sum, right_sum, middle_sum` from `num_int` as

```
from num_int import *
from my_funs import g
```

and evaluate the integral (b). Uncomment the exact value of the integral $g(x)$, replace the function f by g in the calling sequence and run the code and compare the numerical approximation results. Which method is better?

```
approx1=left_sum(g,a,b,n)
approx2=right_sum(g,a,b,n)
approx3=middle_sum(g,a,b,n)
```

6. You can also get the information about python modules and the functions inside the module using the python function `help()`. In Python or (**Spyder**) IPython console type:

```
$ cd ~/MA305/Classwork/CW6
$ python                                                    (or ipython)
>>> import num_int
>>> help (num_int)
>>> num_int.left_sum.__doc__
```

7. Let us try to evaluate the integral (a) in python console using the function `middle_sum` from the module `num_int`.

```
$ python                                                    (or ipython)
>>> from num_int import middle_sum as ms
>>> f = lambda x:  (2*x+1)*(x**2+x+1)**3
>>> ms(f,0,1,11)
>>> ms(f,0,1,101)
```

8. Prepare a typescript file showing your code **cw6a.py** and its execution, rename it as **yourname_cw6script.txt**, make it readable (clean it up!), and zip all the files to one.

```
$ zip yourname_cw6.zip cw6a.py my_funs.py num_int.py yourname_cw6script.txt
```

---

9. Submit the file **yourname_cw6.zip** through Canvas. **No email!**