1 Task 1

I choose to use Python Programming Language for this class.

- The code was complied using PyCharm;
- The compilation resulted with an executable **print**.

```
import numpy as np
import math

print("Hello, world!")
print("Response to 'it's only a bunny': JESUS CHRIST")
```

It has the following output:

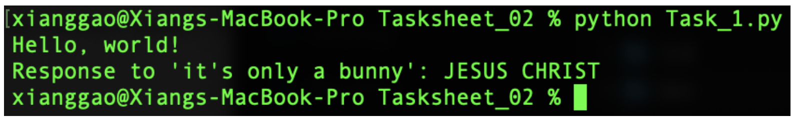

Figure 1. Code Output.

2 Task 2

I have edited my main **README.md** file to include an introduction for the repository. There's now also a table of contents for the homework problems and a link to the software manual I will create.
The file can be found by clicking here.

3 Task 3

The Taylor series expansion of a function $f(a+h)$ centered at $x = a$ is given by

$$f(a+h) = f(a) + hf'(a) + h^2\frac{f''(a)}{2!} + h^3\frac{f'''(a)}{3!} + \dots$$

If we don't want to torture yourself, we can approximate it as

$$\begin{aligned} f(a+b) &= f(a) + hf'(a) \\ \implies f'(a) &= \frac{f(a+h) - f(a)}{h} \end{aligned}$$

Now, to use the centered difference approximation for the first derivative, which is given by

$$\begin{aligned} f(a) &= \frac{f(a+h) - f(a-h)}{2h} \\ &= \frac{1}{2h}\left[\left(f(a) + f'(a)(h) + \frac{1}{2}f''(a)h^2 + \frac{1}{6}f'''(\xi_1)(h^3\right) + \left(f(a) + f'(a)(-h) + \frac{1}{2}f''(a)(-h)^2 + \frac{1}{6}f'''(\xi_1)(-h)^3\right)\right] \\ &= \frac{1}{2h}\left(2f'(a)h + \frac{1}{6}\left(f'''(\xi_1) + f'''(\xi_2)\right)h^2\right) \\ &= f'(a) + \frac{1}{12}h^2 f'''(\xi_3) \end{aligned}$$

And we can rearrange to get:

$$\frac{f(a+h) - f(a-h)}{2h} - f'(a) = \frac{1}{12}h^2 f'''(\xi_3)$$

which suggests that the order of accuracy with h^2, that's why the centered difference approximation isa second order approximation.

4 Task 4

The code is the following:

```python
import numpy as np
import math

np.set_printoptions(precision=24)

h = np.zeros(18)
h[0] = 1
h[1] = 0.5

for i in range(2, 18):
    h[i] = math.pow(10, -(i-1))

#print(h)

A = np.zeros([17, 4])
# initialize the Matrix A

for i in range(0, 17):
    x = 2
    A[i][0] = h[i]
    A[i][1] = -np.cos(2)
    A[i][2] = (np.cos(x + h[i]) - 2 * (np.cos(x)) + np.cos(x - h[i])) / (math.pow(h[i], 2))
    A[i][3] = np.abs(A[i][2] - A[i][1])

# print(A)

print('{0:<36s}{1:<26s}{2:<36s}{3:<24s}'.format('h-value', 'Exact', 'Approximation', 'Difference'))

for i in range(0, 17):
    print('{0:.16f}{1:<12s}{2:.16f}{3:<12s}{4:.16f}{5:<19s}{6:.16f}'.format(A[i][0], "  ", A[i][1], "  ", A[i][2],
                                                                          "   ", A[i][3]))
```

With the following output:

```
xianggao@Xiangs-MacBook-Pro Tasksheet_02 % python Task_4.py
h-value                             Exact                     Approximation                       Difference
1.0000000000000000            0.4161468365471424            0.3826034823619792                  0.0335433541851632
0.5000000000000000            0.4161468365471424            0.4075490368602161                  0.0085977996869263
0.1000000000000000            0.4161468365471424            0.4158001630923890                  0.0003466734547534
0.0100000000000000            0.4161468365471424            0.4161433686711291                  0.0000034678760133
0.0010000000000000            0.4161468365471424            0.4161468019070469                  0.0000000346400955
0.0001000000000000            0.4161468365471424            0.4161468170060800                  0.0000000195410624
0.0000100000000000            0.4161468365471424            0.4161471167662966                  0.0000002802191542
0.0000010000000000            0.4161468365471424            0.4160005673270462                  0.0001462692200963
0.0000001000000000            0.4161468365471424            0.4385380947269369                  0.0223912581797945
0.0000000100000000            0.4161468365471424            1.1102230246251563                  0.6940761880780140
0.0000000010000000            0.4161468365471424            55.5111512312578199                  55.0950043947106778
0.0000000001000000            0.4161468365471424            5551.1151231257817926                  5550.6989762892344515
0.0000000000100000            0.4161468365471424            555111.5123125782702118                  555111.0961657416773960
0.0000000000010000            0.4161468365471424            0.0000000000000000                  0.4161468365471424
0.0000000000001000            0.4161468365471424            5551115123.1257820129394531                  5551115122.7096347808837891
0.0000000000000100            0.4161468365471424            -1665334536937.7348632812500000                  1665334536938.1511230468750000
0.0000000000000010            0.4161468365471424            277555756156289.1250000000000000                  277555756156288.7187500000000000
```

Figure 2. Code Output.

5 Task 5

There are three finite difference approximations documented here[1].

- One-sided approximation
    1. Forward difference approximation

$$\frac{f(x+h)-f(x)}{h} = \frac{f(x+\delta x)-f(x)}{\delta x}$$

[1] https://archive.siam.org/books/ot98/sample/OT98Chapter1.pdf

2. Backward difference approximation

$$\frac{f(x+h)-f(x)}{h} = \frac{f(x)-f(x-\delta x)}{\delta x}$$

- Centered approximation

$$\frac{f(x+\delta x)-f(x-\delta x)}{2\delta x} - f'(x) = \delta x^2 \frac{1}{12} f'''(\xi_3)$$

Each of the given example above is found here[2].
In science and engineering, we often don't know the exact formula for $f(x)$, and given a set of experimental data if we want to know the rate of change of $f(x)$ with respect to x, our best choice would be using the finite approximation methods.

[2]https://www.dam.brown.edu/people/alcyew/handouts/numdiff.pdf