Ngày .... Tháng ..... Năm.........

Nguyễn Minh Đức - ITITIU21045

Q1

a) Let $f(x) = e^x$, so we have $f'(x) = f''(x) = f^{(3)}(x) = f^{(n)}(x) = e^x$

We apply Taylor's series:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \dots + \frac{f^{(n)}(x_i)}{n!}h^n$$

$$\rightarrow e^{x_{i+1}} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

b) Let $f(x) = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$

$$g(x) = \frac{1}{1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots}$$

~~Approaches / Order | f(x) | TRE (true) | ARE (approximation) | g(x) | TRE | ARE~~

~~0th~~

True value = 0.049787

Approaches / Order	f(x)	TRE	ARE	Compare w/ true value	g(x)	TRE	ARE	Compare w/ true value
0^{th}	1	-1908.56%		>	1	-1908.56%		>
1^{st}	-2	4117.11%	150%	<	0.25	-402.14%	-300%	>
2^{nd}	2.5	-4921.39%	180%	>	~0.118	-137%	-111.86%	>
3^{rd}	-2	4117.11%	225%	<	~0.077	-54.66%	-53.25%	>
4^{th}	1.375	-2661.77%	245.45%	>	~0.077	-54.66%	-25.96%	>

Q2. • First derivative Choose $x = 0.25 \to f'(0.25) = -1.8125$

Forward: $f'(0.25) = \dfrac{f(x+h) - f(x)}{h} = \dfrac{f(0.5) - f(0.25)}{0.25} = \cancel{-1.9375}\ -1.5625$

Backward: $f'(0.25) = \dfrac{f(x) - f(x-h)}{h} = \dfrac{f(0.25) - f(0)}{0.25} = -1.9375$

Center: $f'(0.25) = \dfrac{f(x+h) - f(x-h)}{2h} = \dfrac{f(0.5) - f(0)}{2 \times 0.25} = \cancel{-1.9375}\ -1.75$

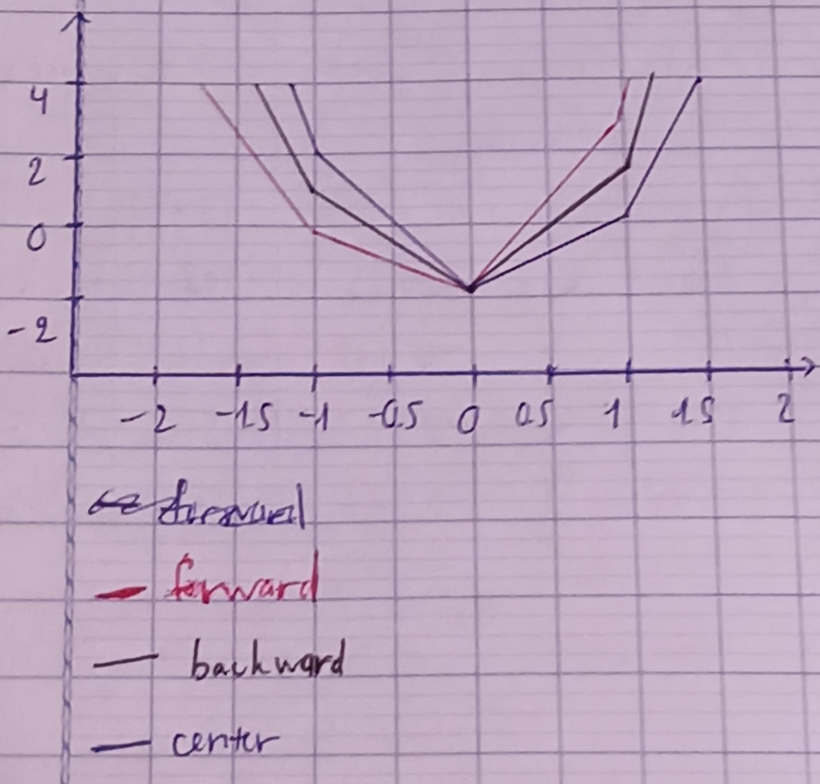
• Second derivative $f'(x) = 3x^2 - 2$

Forward: $f''(0.25) = \dfrac{f'(x+h) - f'(x)}{h} = \dfrac{f'(0.5) - f'(0.25)}{0.25} = 2.25$

Backward: $f''(0.25) = \dfrac{f'(x) - f'(x-h)}{h} = \dfrac{f'(0.25) - f'(0)}{0.25} = 0.75$

Center: $f''(0.25) = \dfrac{f(x+h) - 2f(x) + f(x-h)}{h^2} = \dfrac{f(0.5) - 2f(0.25) + f(0)}{0.25^2} = 1.5$

First derivative:



Q3: $H = A_e \sigma T^4 = 4\pi r^2 e \sigma T^4$

Given data:

$\tilde{r} = 0.15 \qquad \Delta\tilde{r} = 0.02$

$\tilde{e} = 0.9 \qquad \Delta\tilde{e} = 0.04$

$\tilde{T} = 550 \qquad \Delta\tilde{T} = 17.5$

We have

$$\Delta H(\tilde{r}, \tilde{e}, \tilde{T}) = \left|\frac{dH}{dr}\right| \Delta\tilde{r} + \left|\frac{dH}{de}\right| \Delta\tilde{e} + \left|\frac{dH}{dT}\right| \Delta\tilde{T}$$

$$= 8\pi r e \sigma T^4 . \Delta\tilde{r} + 4\pi r^2 \sigma T^4 \Delta\tilde{e} + 16\pi r^2 e \sigma T^3 . \Delta\tilde{T}$$

$$= 578.7929$$

$$H(max) = 4\pi r_{max}^2 . e_{max} . T_{max}^4 \sigma$$

$$= 2007.62$$

$$H(actual) = 4\pi \tilde{r}^2 \tilde{e} \tilde{T}^4 \sigma$$

$$= 1320.29$$

True error $= \cancel{f(562} \; H(max) - H(actual)$

$= 2007.62 - 1320.29 =$

$= 687.33$

$\Rightarrow$ The true error is larger than the estimated error

Q4. a) $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$, Choose $x_i = 0$; $h = x$

Let $f(x) = \sin x \rightarrow f'(x) = \cos x$; $f''(x) = -\sin x$; $f^{(3)}(x) = -\cos x$;

$f^{(4)}(x) = \sin x, \ldots$

Taylor's series:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n$$

$$\sin x_{i+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \quad (4 \text{ terms})$$

b) $x_{i+1} = \frac{\pi}{6}$; $\sin\frac{\pi}{6} = 0.5$

• We try with 4 terms:

$$\sin\left(\frac{\pi}{6}\right) = \frac{\pi}{6} - \frac{\left(\frac{\pi}{6}\right)^3}{3!} + \frac{\left(\frac{\pi}{6}\right)^5}{5!} - \frac{\left(\frac{\pi}{6}\right)^7}{7!} = 0.4999999919$$

$$\text{True error} = 0.5 - 0.4999999919 = 0.\underbrace{00000008}_{9\text{ sig. figures}}1$$

• 3 terms

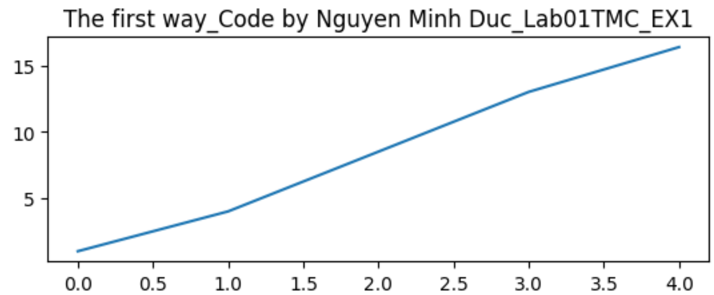
$$\sin\left(\frac{\pi}{6}\right) = \frac{\pi}{6} - \frac{\left(\frac{\pi}{6}\right)^3}{3!} + \frac{\left(\frac{\pi}{6}\right)^5}{5!} = 0.5000021326$$

$$\text{True error} = 0.5 - 0.5000021326 = -0.\underbrace{000002}_{6\text{ sig. figures}}132$$

Therefore, we need 3 terms to have 6 significant figures

```
import math
import numpy as np
import matplotlib.pyplot as p


real_value = np.exp(-3)


l1 = []
l2 = []


def firstway(x, term):
  result = 0
  prev = 0
  e_t = 0
  e_a = 0
  print("Calculation using first way: ")
  print("%s%20s%20s%20s" % ("Iteration", "Approximation", "E_a", "E_t"))
  for n in range(0, term):
    prev = result
    result += math.pow(-x, n) / math.factorial(n)
    l1.append(result)

    e_t = ((real_value - result) / real_value)
    e_a = ((result - prev) / result)
    print("%9d%20.10f%20.10f%20.10f" % (n, result, e_a, e_t))


def secondway(x, term):
  result = 0
  prev = 0
  e_t = 0
  e_a = 0
  print("Calculation using second way: ")
  print("%s%20s%20s%20s" % ("Iteration", "Approximation", "E_a", "E_t"))
  for n in range(0, term):
    prev = result
    if result != 0:
      flip_result = 1/result
    else:
      flip_result = 0
    flip_result += math.pow(x, n) / math.factorial(n)
    result = 1/flip_result
    l2.append(result)

    e_t = ((real_value - result) / real_value)
    e_a = ((result - prev) / result)
    print("%9d%20.10f%20.10f%20.10f" % (n, result, e_a, e_t))


firstway(-3,5)
print("\n")
secondway(-3,5)
p.subplot(2,1,1)
p.plot(l1)
p.title("\nThe first way_Code by Nguyen Minh Duc_Lab01TMC_EX1")
```

```
Calculation using first way:
Iteration       Approximation                E_a                E_t
        0        1.0000000000       1.0000000000      -19.0855369232
        1        4.0000000000       0.7500000000      -79.3421476928
        2        8.5000000000       0.5294117647     -169.7270638471
        3       13.0000000000       0.3461538462     -260.1119800014
        4       16.3750000000       0.2061068702     -327.9006671172


Calculation using second way:
Iteration       Approximation                E_a                E_t
        0        1.0000000000       1.0000000000      -19.0855369232
        1       -0.5000000000       3.0000000000       11.0427684616
        2        0.4000000000       2.2500000000       -7.0342147693
        3       -0.5000000000       1.8000000000       11.0427684616
        4        0.7272727273       1.6875000000      -13.6076632169
Text(0.5, 1.0, '\nThe first way_Code by Nguyen Minh Duc_Lab01TMC_EX1')
```



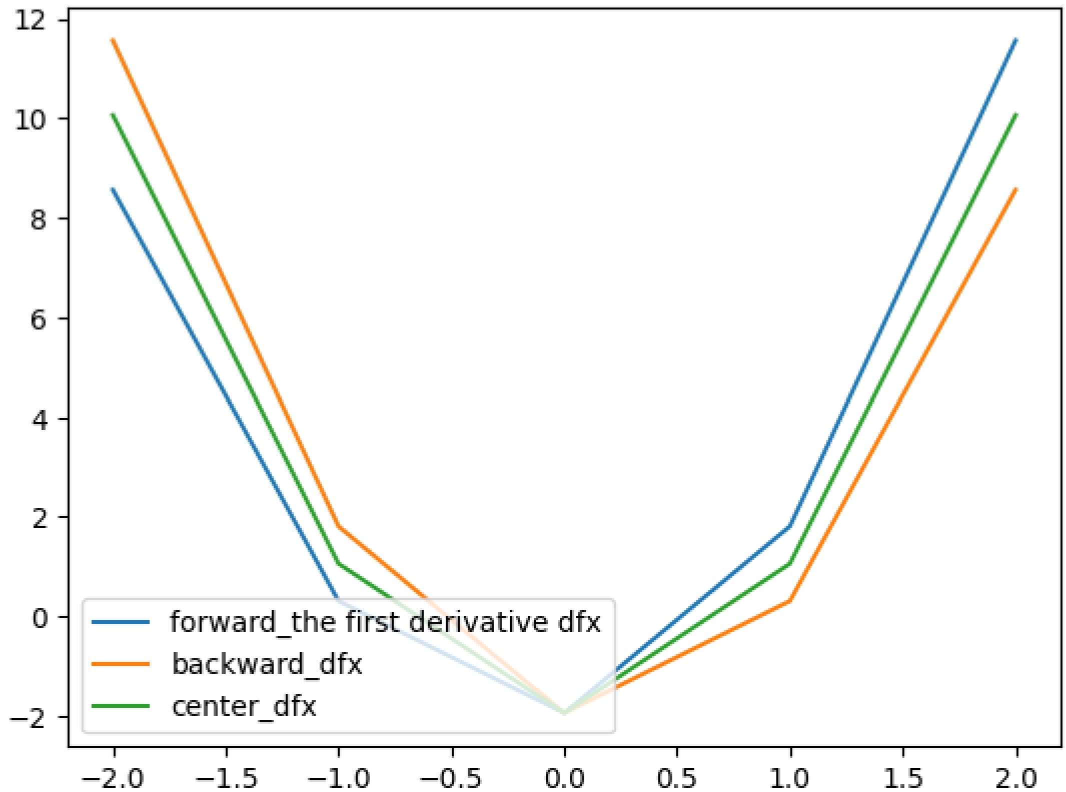
```
#Code by Nguyen Minh Duc_ITITIU21045_Lab01TMC_EX2
import math, numpy as np, matplotlib.pyplot as plt


fx=lambda x: x**3 - 2*x + 4


dfx = lambda x: 3*x**2 - 2


array1_dfx=[]
array2_dfx=[]
array3_dfx=[]
array1_d2fx=[]
array2_d2fx=[]
array3_d2fx=[]


interval=[-2,-1,0,1,2]


def forward(x,h,f):
  return (f(x+h)-f(x))/h
```

+ Code + Text

```
def backward(x,h,f):
  return (f(x)-f(x-h))/h


def center_first(x,h,f):
  f_n=f(x+h)
  f_p=f(x-h)
  z=2*h
  return (f_n-f_p)/z


def center_second(x,h,f):
  f_n=f(x+h)
  f_p=f(x-h)
  z=h**2
  return (f_n-2*f(x)+f_p)/z


for i in range(-2,3):
  array1_dfx.append(forward(i,0.25,fx))
  array2_dfx.append(backward(i,0.25,fx))
  array3_dfx.append(center_first(i,0.25,fx))
  array1_d2fx.append(forward(i,0.25,dfx))
  array2_d2fx.append(backward(i,0.25,dfx))
  array3_d2fx.append(center_second(i,0.25,dfx))
```

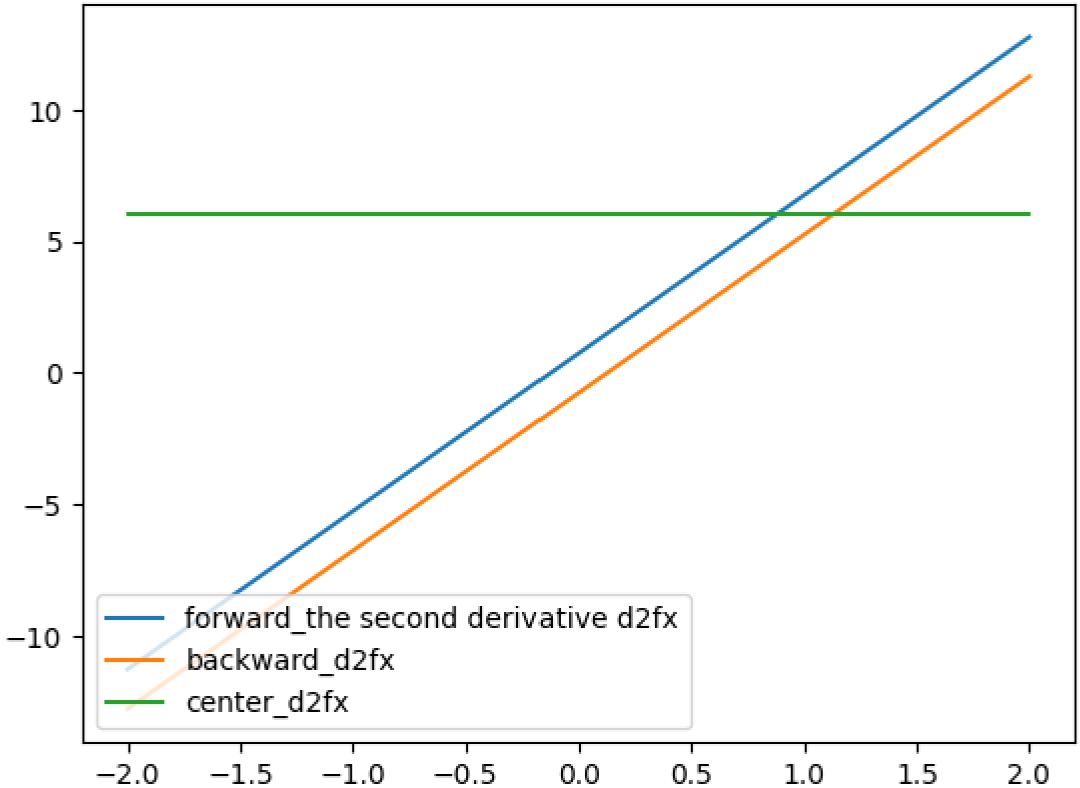
```
plt.plot(interval,array1_dfx)
plt.plot(interval,array2_dfx)
plt.plot(interval,array3_dfx)
plt.legend(['forward_the first derivative dfx','backward_dfx','center_dfx'],loc="lower left")
```

```
<matplotlib.legend.Legend at 0x7be229df4370>
```




```
plt.plot(interval,array1_d2fx)
plt.plot(interval,array2_d2fx)
plt.plot(interval,array3_d2fx)
plt.legend(['forward_the second derivative d2fx','backward_d2fx','center_d2fx'],l
```

```
<matplotlib.legend.Legend at 0x7be229d8f040>
```



```
#Code by Nguyen Minh Duc_ITITIU21045_Lab01TMC_EX2


import math
def maclaurin_sin(x,n):
  """
  Calculates the Maclaurin series expansion of sin(x) up to n terms
  """
  result=0
  for i in range(n):
    sign=(-1)**i
    term=x**(2*i+1)/math.factorial(2*i+1)
    result += sign*term
  return result


def taylor_sin(x,n):
  """
  Calculates the Taylor series expansion of sin(x) up to n terms
  """
  result=0
  for i in range(n):
    sign=(-1)**i
    term=x**(2*i+1)/math.factorial(2*i+1)
    result += sign*term
  return result


n=1
x=math.pi/6
eps=0.5*10**(-6)
while abs(maclaurin_sin(x,n)-math.sin(x))>=eps:
  n+=1
print("Maclaurin series expansion: ")
print(f"sin({x})={maclaurin_sin(x,n)} (approximation using {n} terms)")
```

```
Maclaurin series expansion:
sin(0.5235987755982988)=0.4999999918690232 (approximation using 4 terms)
```