Abu Butt March 2, 2017

CSc 30100

Assignment 1

1.

Abu Butt 3/1/17

Csc 301

1 a) Derive $f_2(x, \delta)$.

$f_1(x,\delta) = \cos(x+\delta) - \cos(x)$ can be transformed into

$$\cos(\phi) - \cos(\psi) = -2\sin\left(\frac{\phi+\psi}{2}\right)\sin\left(\frac{\phi-\psi}{2}\right)$$

$\phi = x+\delta$

$\psi = x$

$$= -2\sin\left(\frac{\phi+\psi}{2}\right)\sin\left(\frac{\phi-\psi}{2}\right)$$

$$= -2\sin\left(\frac{x+\delta+x}{2}\right)\sin\left(\frac{x+\delta-x}{2}\right)$$

$$= -2\sin\left(\frac{2x+\delta}{2}\right)\sin\left(\frac{\delta}{2}\right)$$

$$\boxed{f_2(x,\delta)} \boxed{= -2\sin\left(x+\frac{\delta}{2}\right)\sin\left(\frac{\delta}{2}\right)}$$

1(b) $g_1(x,\delta) = f_1(x,\delta)/\delta + \sin(x)$

$$\boxed{= \frac{\cos(x+\delta) - \cos(x)}{\delta + \sin(x)}}$$

$g_2(x,\delta) = f_2(x,\delta)/\delta + \sin(x)$

$$= \boxed{\frac{-2\sin\left(x+\frac{\delta}{2}\right)\sin\left(\frac{\delta}{2}\right)}{\delta + \sin(x)}}$$

c) Explain the difference in the results of the two calculations.

- There was a small difference between both function's calculations. f1 and f2 should have the same values since both of them are equal. g1 and g2 has different value due to inaccuracy between f1 and f2. Both functions f1 and f2 were divided by small value of delta(δ), which result in errors in any calculations.

Code for Question 1:

ex1b.py

```python
import math


delta = 1.e-11;
x = 3;

f1 = math.cos(x+delta) - math.cos(x);
f2 = -2 * math.sin(x + (delta/2)) * math.sin(delta/2);

g1 = f1 / delta + math.sin(x);
g2 = f2 / delta + math.sin(x);

print "f1 : {} ".format(f1);
print "f2 : {} ".format(f2);
print "g1 : {} ".format(g1);
print "g2 : {} ".format(g2);
```

Desktop — -bash — 80×24

```
Abus-MBP:Desktop ABUBUTT1$ python ex1b.py
f1 : -1.4112044866e-12
f2 : -1.41120008055e-12
g1 : -4.40600236434e-07
g2 : 4.94995711087e-12
Abus-MBP:Desktop ABUBUTT1$
```

2.

2) $f_1(x_0, h) = \sin(x_0+h) - \sin(x_0)$ into $f_2(x_0, h)$

$$\sin(\phi) - \sin(\psi) = 2\cos\left(\frac{\phi+\psi}{2}\right)\sin\left(\frac{\phi-\psi}{2}\right)$$

a) Derive $f_2(x_0, h)$

$\phi = x_0 + h$

$\psi = x_0$

$$f_1(x_0, h) = \sin(x_0+h) - \sin(x_0)$$

$$f_2(x_0, h) = 2\cos\left(\frac{\phi+\psi}{2}\right)\sin\left(\frac{\phi-\psi}{2}\right)$$
$$= 2\cos\left(\frac{x_0+h+x_0}{2}\right)\sin\left(\frac{x_0+h-x_0}{2}\right)$$
$$= 2\cos\left(\frac{2x_0+h}{2}\right)\sin\left(\frac{h}{2}\right)$$
$$= \boxed{2\cos\left(x_0+\frac{h}{2}\right)\sin\left(\frac{h}{2}\right)}$$

2b)

$$f'(x) = \frac{f(x_0+h) - f(x_0)}{h}$$

Since $\frac{d}{dx}\sin(x) = \lim_{h \to 0} \frac{\sin(x+h) - \sin(x)}{h}$

$$f'(x) = \frac{\sin(x_0+h) - \sin(x_0)}{h}$$

as h goes to zero, f1 doesn't have any values, therefore we need a formula for cancellation errors.

so f'(x) becomes

$$f'(x) = \frac{\sin(x_0+h) - \sin(x_0)}{h} \rightarrow f'(x) = \frac{2\cos\left(x_0+\frac{h}{2}\right)\sin\left(\frac{h}{2}\right)}{h}$$

when we plug new f'(x) formula, we get more accurate results as h approaches zeros.

3. the given points are mainly distributed close to zero.

rounding unit:

$$\eta = \frac{1}{2}\beta^{1-t}$$

$\beta = 2$

$t = 3$

$$\eta = \frac{1}{2} 2^{1-3}$$

$$\eta = \frac{1}{2} 2^{-2}$$

$$\eta = \left(\frac{1}{2}\right)\left(\frac{1}{4}\right)$$

$$\boxed{\eta = \frac{1}{8} / 0.125}$$

Code for Question 2:

2b.py

```python
import math

x = 1.2;
h = 1.e-20;


for i in range(20,0,-1):
    f1 = math.sin(x+h) - math.sin(x);
    f2 = f1/h;
    print("h: 1.e-%i " %i) + ' f1 = ' + '{0:.18f}'.format(f1);
    h = h * 10.0;
```

Desktop — -bash — 80×24

```
Abus-MacBook-Pro:Desktop ABUBUTT1$ python 2b.py
h: 1.e-20  f1 = 0.000000000000000000
h: 1.e-19  f1 = 0.000000000000000000
h: 1.e-18  f1 = 0.000000000000000000
h: 1.e-17  f1 = 0.000000000000000000
h: 1.e-16  f1 = 0.000000000000000000
h: 1.e-15  f1 = 0.000000000000000444
h: 1.e-14  f1 = 0.000000000000003664
h: 1.e-13  f1 = 0.000000000000036193
h: 1.e-12  f1 = 0.000000000000362488
h: 1.e-11  f1 = 0.000000000003623657
h: 1.e-10  f1 = 0.000000000036235792
h: 1.e-9  f1 = 0.000000000362357810
h: 1.e-8  f1 = 0.000000003623577549
h: 1.e-7  f1 = 0.000000036235770828
h: 1.e-6  f1 = 0.000000362357288508
h: 1.e-5  f1 = 0.000003623530942853
h: 1.e-4  f1 = 0.000036231115191909
h: 1.e-3  f1 = 0.000361891674579562
h: 1.e-2  f1 = 0.003576915586159579
h: 1.e-1  f1 = 0.031519099449966670
Abus-MacBook-Pro:Desktop ABUBUTT1$
```

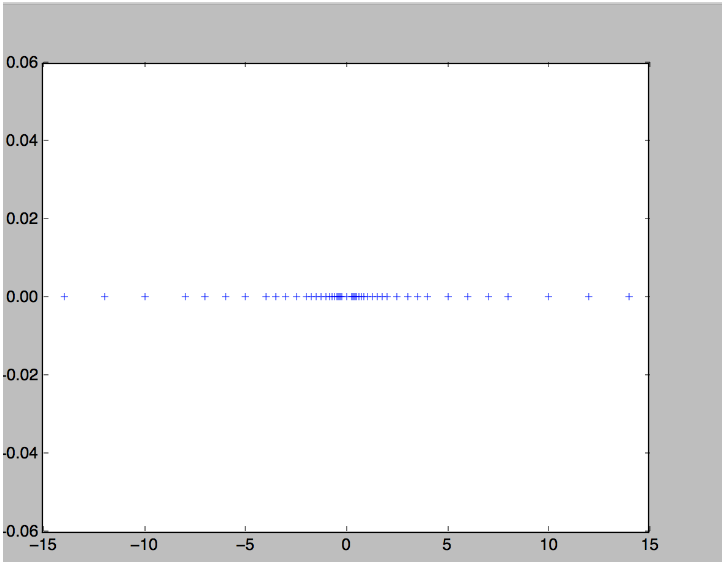
```python
x = 1.2;
h = 1.e-20;


for i in range(20,0,-1):
    f_prime = 2*math.cos(x+h/2) * math.sin(h/2);
    f1_prime = f_prime/h;
    print("h: 1.e-%i " %i) + ' f_prime = ' + '{0:.21f}'.format(f1_prime);
    h = h * 10.0;
```

Desktop — -bash — 80×24

```
Abus-MacBook-Pro:Desktop ABUBUTT1$ python 2b.py
h: 1.e-20  f_prime = 0.362357754476673621369
h: 1.e-19  f_prime = 0.362357754476673621369
h: 1.e-18  f_prime = 0.362357754476673621369
h: 1.e-17  f_prime = 0.362357754476673565858
h: 1.e-16  f_prime = 0.362357754476673621369
h: 1.e-15  f_prime = 0.362357754476673177280
h: 1.e-14  f_prime = 0.362357754476668847410
h: 1.e-13  f_prime = 0.362357754476627047513
h: 1.e-12  f_prime = 0.362357754476207549743
h: 1.e-11  f_prime = 0.362357754472013404712
h: 1.e-10  f_prime = 0.362357754430071676843
h: 1.e-9  f_prime = 0.362357754010654009580
h: 1.e-8  f_prime = 0.362357749816478225124
h: 1.e-7  f_prime = 0.362357707874718826258
h: 1.e-6  f_prime = 0.362357288457070159104
h: 1.e-5  f_prime = 0.362353094275204457020
h: 1.e-4  f_prime = 0.362311151918484397605
h: 1.e-3  f_prime = 0.361891674579568889403
h: 1.e-2  f_prime = 0.357691558615958526968
h: 1.e-1  f_prime = 0.315190994499666143902
Abus-MacBook-Pro:Desktop ABUBUTT1$
```

As you can see the difference between both formula and calculation results of them. In the second formula, I have more accurate results than the first formula. Second formula, there is no subtraction, only multiplication which is excluding the possibility if any error.

Code for Question 3:




```
xlist = [];
xlistAll = [];
zerolist = [];


for i in np.arange(1.0,2.0,0.25):
    for j in range(-2,4,1):
        x = i * (2**j);
        xlist.append(x);
for i in xlist:
    xlistAll.append(i * -1.0);

xlistAll.append(0);

xlistAll = xlistAll + xlist;

xlistAll.sort();

for i in xlistAll:
    zerolist.append(0.0);
plt.plot(xlistAll,zerolist, '+');
plt.show();
```

As you can see the points are distributed close to zero both axes (0,0.0). Mantissa is 3, and exponents are from -2 to 3.

4.

4) $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$ as n get large

$$S^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2$$

a) Method 2 is cheaper in terms of flop counts. Because there are less operations to perform than method 1. For method, we subtract everytime when there is an operation. Where as in Method 2, we subtract only once in the end.

b) Method 2 will be more accurate in terms of results because subtraction happen when the summation is completed. where as in method 1, we subtact every time, which ends in losing some accuracy in the results.