Problem Set 1

1. In final plot.

2. The FD-1 approximation first displays larger, sporadic error which decreases until about $\Delta x = 10^{-8.5}$, when the error in the approximation begins to grow linearly. The smallest error in $f'(x0)$ is about 10^{-9}, and it occurs at approximately $\Delta x = 10^{-8.5}$. This is the effect of the roundoff error diminishing as the dominant source of error in the estimate, and the discretionary error taking over, as can be seen when the discretionary error is added into the plot.

3.

$$f(x0 + \Delta x) = f(x0) + \Delta x f'(x0) + \frac{\Delta x^2}{2} f''(x0) + \cdots$$

$$f'(x0) = \frac{f(x0 + \Delta x) - f(x0)}{\Delta x} + \frac{\Delta x}{2} f''(x0) + \cdots$$

Discretionary error D is approximated as the first neglected term, $D \approx \frac{\Delta x}{2} f'\prime(x0)$.

4. $f(x0 + \Delta x)$ can be seen above,

$$f(x - \Delta x) = f(x0) - \Delta x f'(x0) + \frac{\Delta x^2}{2} f''(x0) + \cdots$$

$$f'_{FD-2}(x0) = \frac{f(x0 + \Delta x) - f(x0 - \Delta x)}{2\Delta x}$$

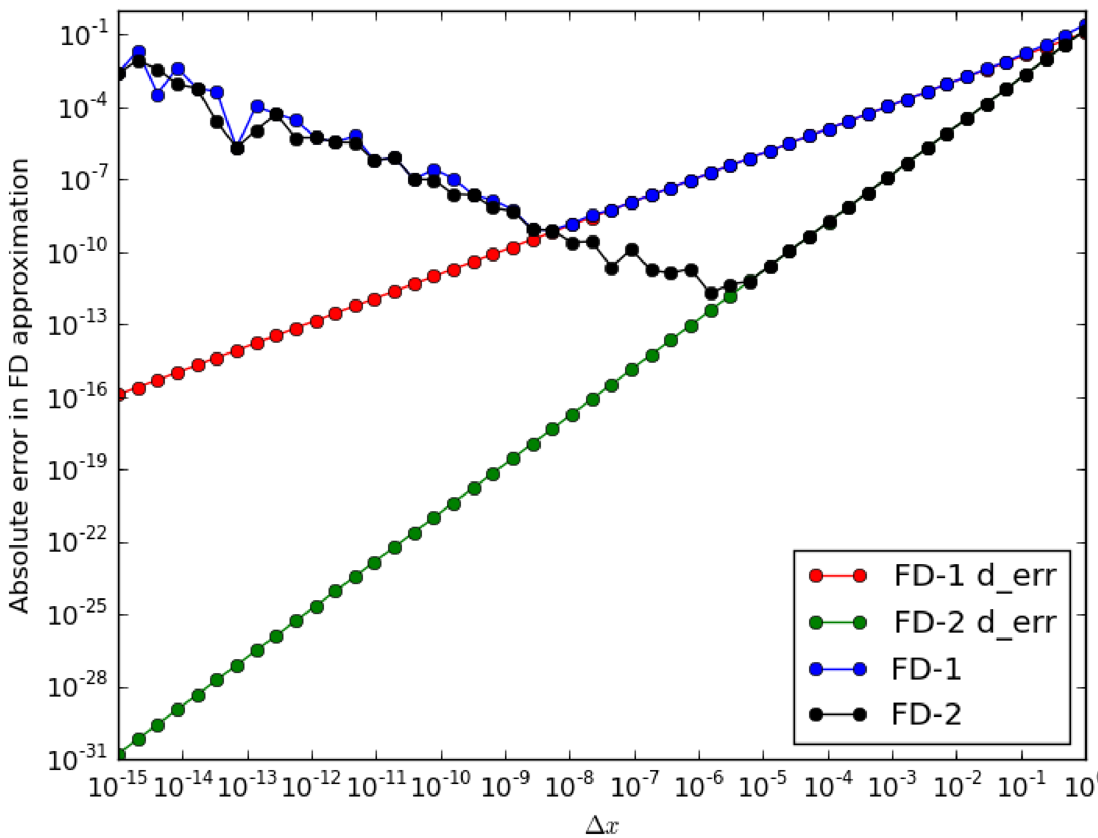
$$= \frac{1}{2\Delta x}(f(x0) + \Delta x f'(x0) + \frac{\Delta x^2}{2} f''(x0) + \cdots - (f(x0) - \Delta x f'(x0) + \frac{\Delta x^2}{2} f''(x0) + \cdots))$$

$$= f'(x0) + \frac{\Delta x^2}{6} f'''(x0) + \cdots$$

FD-2 is special in comparison to FD-1 because it removes all terms even in powers of Δx from the Taylor expansion, resulting in a smaller discretionary error, $D \approx \frac{\Delta x^2}{6} f'''(x0)$

5. The FD-2 approximation behaves similarly to the FD-1, but with a consistently lower absolute error.

6. FD-1 is a first order approximation because we lose all terms after the first derivative in a Taylor expansion of f, while FD-2 is a second order approximation because we lose all terms after the second derivative in a Taylor expansion of f. If we want the lowest error at the lowest computational cost, FD-2 is better to use because we can use a larger Δx and achive a lower error than with FD-1.


Absolute error in FD approximation
10^{-1}
10^{-4}
10^{-7}
10^{-10}
10^{-13}
10^{-16}
10^{-19}
10^{-22}
10^{-25}
10^{-28}
10^{-31}
10^{-15} 10^{-14} 10^{-13} 10^{-12} 10^{-11} 10^{-10} 10^{-9} 10^{-8} 10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{0}
Δx
FD-1 d_err
FD-2 d_err
FD-1
FD-2

```python
#!/usr/bin/python3.4

import numpy as np
from matplotlib import pyplot as plt

def first_order_finite_diff(f, x, delta_x):
    return (f(x+delta_x)-f(x))/delta_x

def second_order_finite_diff(f, x, delta_x):
    return (f(x+delta_x)-f(x-delta_x))/(2*delta_x)


def plot_FD_error(f, dfdx, x0, ax, min_delta_x=-10, max_delta_x=10, ddfdxx=None, dddfdxxx=None) :
    """Function will find and plot the error in the FD approximation """

    #Finding the error in the FD approximation
    delta_x = np.logspace(min_delta_x, max_delta_x, base = 10.0)

    FD_1_error = np.abs(first_order_finite_diff(f, x0, delta_x)-dfdx(x0))
    #print(FD_1_error) Default code

    FD_2_error = np.abs(second_order_finite_diff(f, x0, delta_x)-dfdx(x0))


    #Setting up the plot
    #ax.loglog(delta_x, FD_1_error, label='FD-1') Default code
    if ddfdxx != None:
        d_err = np.abs(delta_x/2.*ddfdxx(x0))
        ax.loglog(delta_x, d_err, 'r-o', label='FD-1 d_err')
    if dddfdxxx != None:
        d_err2 = np.abs(delta_x**2/6.*dddfdxxx(x0))
        ax.loglog(delta_x, d_err2, 'g-o', label='FD-2 d_err')
    ax.loglog(delta_x, FD_1_error, 'b-o', label='FD-1')
    ax.loglog(delta_x, FD_2_error, 'k-o', label='FD-2')
    ax.set_xlabel(r'$\Delta x$')
    ax.set_ylabel('Absolute error in FD approximation')
    ax.legend(loc="lower right")



if __name__=="__main__":

    figure_1 = plt.figure()
    ax = figure_1.add_subplot(1,1,1)

    #Defining the particulars of the function to be approximated
    f = np.sin
    dfdx = np.cos
```

```
    ddfdxx = lambda x: -1*np.sin(x)
    dddfdxxx = lambda x: -1*np.cos(x)
    x0 = .25

    #plot_FD_error(f, dfdx, x0, ax) Default code
    #plot_FD_error(f, dfdx, x0, ax, -20,1) Problem 2
    #plot_FD_error(f, dfdx, x0, ax, -20,1, ddfdxx) Problem 3
    #plot_FD_error(f, dfdx, x0, ax, -20,1, ddfdxx, dddfdxxx) Problem 4
    plot_FD_error(f, dfdx, x0, ax, -15,0, ddfdxx, dddfdxxx)
    figure_1.savefig("hw1_fig1.png")
    plt.show()
```