

```

# Matthew Pahayo
# 1/28/2021
# computational methods
# hw 1
# question 1
# newtons method
# q1_nm.py

# create variable
import sympy as sym
x = sym.Symbol('x')

# define function, initial guess, and tol
f = 2*sym.sin(x**2)-3*sym.sin(x**2)+x**2*sym.sin(x**2)
t_0 = 2.2
tol = 10**-8

# lambdify funtion and its derivative
fp = sym.diff(f)
f = sym.lambdify(x, f)
fp = sym.lambdify(x, fp)

# get next iteration for t and declare lists
t_1 = t_0 - f(t_0)/fp(t_0)
t1 = [t_0, t_1]
f1 = []
dt1 = []

# newtons method
n = 0
while (abs(t1[n+1]-t1[n])/abs(t1[n])) > tol:
    t_np1 = t1[n+1] - f(t1[n+1])/fp(t1[n+1])
    t1.append(t_np1)
    n += 1

# define f1 and dt1
for j in range(len(t1)):
    f1.append(f(t1[j]))
for k in range(len(t1)-1):
    dt1.append(t1[k+1]-t1[k])

data_f1 = open("f1.txt", "a")
data_t1 = open("t1.txt", "a")
for i in range(len(t1)):
    data_t1.write(str(t1[i]) + "\n")
    data_f1.write(str(f1[i]) + "\n")
data_t1.close()
data_f1.close()

i = 0

```

```
data_dtl = open("dtl.txt", "a")
for i in range(len(dtl)):
    data_dtl.write(str(dtl[i]) + "\n")
data_dtl.close()

print("t = " + str(t_np1))
print("total iterations = " + str(n))
print("final delta t = " + str(dtl[n-1]))

# keeps python console up; for non-IDE use
i = 1
while i == 1:
    i = input("Press enter to continue.\n")
```