

- Dry-Eye
- ▼ MasterProject
 - Images
 - PICTURES
 - aa29313-16.pdf
 - ds9
 - firsttry.zip
 - howtouseAARTFAACandTra
 - Notes 19_10_2017
 - Notes_13_10_2017
 - Notes_27_10_1017
 - Notes_27_10_1017~
 - send.zip
 - send2.zip
 - WSCLEANOPTIONS
- Projects_year1_master
- ▼ Projects_year2_master
 - Applied_Machine
 - ▼ Numerical_Algorithms
 - Chapter1
 - ▼ Homework1
 - exc1.pdf
 - exc1.py
 - exc2.pdf
 - exc2.py
 - exc3.py
 - exc3turnin.py
 - homework_assignme
 - Scientific_Computing
 - ▼ UVA_AML17
 - .git
 - src
 - week_1
 - week_2
 - .gitignore
 - LICENSE
 - README.md

Settings
exc2.py
exc3.py
exc3turnin.py

```

5  # Function returning F(x)
6  def Fx(x):
7      return(np.tan(x))
8  # Function returning hat F(x)
9  def AFx(x):
10     a = (x-(1./6)*(x**3))
11     b = (1-((1./2)*(x**2)))
12     return(a/b)
13
14  # Calc Forward error
15  def Forward(x):
16     return(AFx(x)-Fx(x))
17
18  # Calc Backward error
19  def Backward(x):
20     return(np.arctan(AFx(x))-x)
21
22  # Function returning derivative hat F(x)
23  def dAFx(x):
24     return((4 + 0.333333*x**4)/(2 - x**2)**2)
25
26  # Function returning Relative Condition number eval
27  def dRelCond(x):
28     return((x * np.square(1/(np.cos(x)))/Fx(x)))
29
30  # Function returning Relative Condition number eval
31  def adRelCond(x):
32     return((x * dAFx(x))/AFx(x))
33
34  # Print function
35  def Printfunction(x):
36     print("For x = %0.2f" % x)
37     print("-----")
38     print('Forward: %0.2e ' %(Forward(x)))
39     print('Backward: %0.2e ' %(Backward(x)))
40     print('Relative condition evaluating f(x): %0.2
41     print('Relative condition evaluating af(x): %0.
42     print("-----")
43     return 0
44
45  # Initializing values and print functions
46  fx = Fx(1)
47  afx = AFx(1)

```