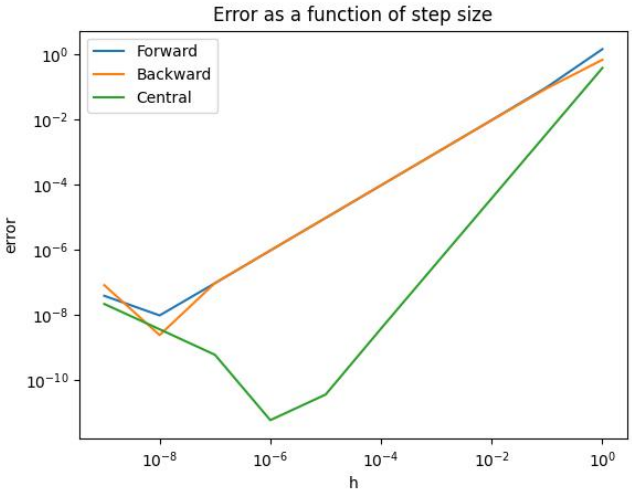
# PYTHON CODE

```
"""
Utilize all three methods of approximation: Forward, Backward, and Central Difference
Initiate the computation with an initial step size of h=1
Successively decrease the step size by a factor of 10 with each iteration.
Plot % Error in log-log scale

"""


from lib.Header import *

def func (x):
    return -.1 * x**4 - .15 * x**3 - .5 * x**2 -.25 * x +1.2

def forwardDiff1(func, x, h):
    return (func(x+h) - func(x)) / h

def backwardDiff1(func, x, h):
    return (func(x) - func(x-h)) / h

def centralDiff1(func, x, h):
    return (func(x+h) - func(x-h)) / (2*h)

x = .5
fpe = -.91250
h = 1
fpte = []
bpte = []
cpte = []
xvalues = []

for ii in np.arange(1, 11):
    fpt = forwardDiff1(func, x, h)
    fpte.append(abs((fpt - fpe)/fpe))
    bpt = backwardDiff1(func, x, h)
    bpte.append(abs((bpt - fpe) / fpe))
    cpt = centralDiff1(func, x, h)
    cpte.append(abs((cpt - fpe) / fpe))
    xvalues.append(h)
    h = h/10


plt.figure()
plt.loglog(xvalues, fpte, label = "Forward")
plt.loglog(xvalues, bpte, label = "Backward")
plt.loglog(xvalues, cpte, label = "Central")
plt.legend()
plt.xlabel("h")
plt.ylabel("error")
```

```
plt.title("Error as a function of step size")
SAVE(1)
plt.show()

PDF()
```

# OUTPUT

## Plots




## Prints