AMSC 460 HW 1 Part 2

Jeffrey Zhang

The goal is to find the root $r = 0$ of $f(x) = x + x^4$ numerically.

a) Apply Newton's method and derive an explicit recursion, expressing the error e_{n+1} in terms of e_n. Predict and report the order of convergence and the asymptotic constant for this problem.

Using the same approach as with Fixed-point Iteration, we can determine the convergence rate of Newton's Method applied to the equation $f(x) = 0$, where we assume that f is continuously differentiable near the exact solution r, and that f''(x) exists for an x near r. Using the Taylor' Theorem, we obtain:

$e_{n+1} = x_{n+1} - r = x_n - \frac{f(x_n)}{f'(x_n)} - r = x_n - r - \frac{f(x_n)}{f'(x_n)} = e_n - \frac{f(x_n)}{f'(x_n)}$. We can redefine $f(x_n)$ using Taylor's Theorem, Thus
$e_{n+1} = e_n - \frac{1}{f'(x_n)}[f(r) - f'(x_n)(r - x_n) - \frac{1}{2}f''(\xi_n)(x_n - r)^2]$. Since $f(r)$ = 0, we can eliminate it in the brackets to obtain
$e_{n+1} = e_n - \frac{1}{f'(x_n)}[-f'(x_n)(r - x_n) - \frac{1}{2}f''(\xi_n)(x_n - r)^2] = e_n$ Since $r - x_n$ is $-e_r$
$+ \frac{1}{f'(x_n)}[f'(x_n)(r - x_n) + \frac{1}{2}f''(\xi_n)(x_n - r)^2]$
$e_{n+1} = e_n - \frac{1}{f'(x_n)}[-f'(x_n)e_n + \frac{1}{2}f''(\xi_n)(x_n - r)^2] = e_n - e_n + \frac{f''(\xi_n)}{2f'(x_n)}e_n^2$
$= \boxed{\frac{f''(\xi_n)}{2f'(x_n)}e_n^2}$

The order of convergence is defined by: $\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \lambda$ where $x_n \neq r \ \forall n$ and α is the order and λ is the asymptotic error constant. We can use the anwer we have above to determine α and λ as follows:

$e_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)} e_n^2 \implies \frac{e_{n+1}}{e_n^2} = \frac{f''(\xi_n)}{2f'(x_n)} \implies \frac{|e_{n+1}|}{|e_n|^2} = \frac{|f''(\xi_n)|}{|2f'(x_n)|} \implies \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^2}$. Hence we can conclude that $\boxed{\alpha = 2}$ and $= \lim_{n\to\infty} \frac{|f''(\xi_n)|}{|2f'(x_n)|} = \frac{|f''(r)|}{|2f'(r)|}$

$\lambda = \frac{|f''(r)|}{|2f'(r)|}$. Thus in the context of this problem with $f'(x) = 1 + 4x^3$, $f''(x) = 12x^2$, and $r = 0$, we have $f'(r) = f'(0) = 1 + 4(0)^3 = 1$ and $f''(r) = f''(0) = 12(0)^2 = 0$. Hence $\lambda = \frac{|0|}{|2 * 1|} = \frac{|0|}{|2|} = 0$. $\boxed{\lambda = 0}$

(b) Implement the Newton iteration. Observe and report how the error decreases over iterations for different initial guesses $x_0 = 0.1, 1, 10$. Is it consistent with you predictions?

In [21]:
```
import math
import numpy as np
from sympy import *
def NewtonsIteration(x0, Tol, N0): #Tol = tolerence, N0 = iterations
    r = 0 #root
    e = x0 - r #error
    x = symbols('x') # x
    f = x + x**4 #function
    df = f.diff(x) #f'(x)
    p0 = x0
    i = 1
    prev_e = e; #previous error
    while i <= N0:
        p = p0 - f.evalf(subs = {x:p0})/df.evalf(subs = {x:p0})
        if np.abs(p - p0) <= Tol:
            print("p = " +str(p) + " after " + str(i) + " iterations.")
            break
        prev_e = e
        p0 = p
        e = p0 - r
        print("error " + str(i)+ ": " + str(e))
        i += 1
        if(i >= 2):
            print("order of convergence 2 for n = " + str(i-1)+ " " + str(e/(prev_e**2)))
    if i > N0:
        print("Method failed after " + str(N0) + " iterations")
```

In [22]:
```
NewtonsIteration(0.1, 10**(-7), 10)
```

```
error 1: 0.000298804780876488
order of convergence 2 for n = 1 0.0298804780876488
error 2: 2.39150062600335e-14
order of convergence 2 for n = 2 2.67852321669054e-7
p = 0 after 3 iterations.
```

In [23]:
```
NewtonsIteration(1, 10**(-7), 10)
```

```
error 1: 0.600000000000000
order of convergence 2 for n = 1 0.600000000000000
error 2: 0.208583690987124
order of convergence 2 for n = 2 0.579399141630901
error 3: 0.00547970709979453
order of convergence 2 for n = 3 0.125949558307823
error 4: 2.70489461905166e-9
order of convergence 2 for n = 4 9.00815103944581e-5
p = 0 after 5 iterations.
```

In [24]:
```
NewtonsIteration(10, 10**(-7), 100)
```

```
error 1: 7.49812546863284
order of convergence 2 for n = 1 0.0749812546863284
error 2: 5.62026107787311
order of convergence 2 for n = 2 0.0999657166165042
error 3: 4.20926823228908
order of convergence 2 for n = 3 0.133258101318045
error 4: 3.14640403462721
order of convergence 2 for n = 4 0.177582958464125
error 5: 2.34101415957187
order of convergence 2 for n = 5 0.236469460169635
error 6: 1.72220140219775
order of convergence 2 for n = 6 0.314250413230143
error 7: 1.23138376797973
order of convergence 2 for n = 7 0.415169601762726
error 8: 0.814483693089217
order of convergence 2 for n = 8 0.537149956314889
error 9: 0.417628863311162
order of convergence 2 for n = 9 0.629543463203442
error 10: 0.0706700815623652
order of convergence 2 for n = 10 0.405186139079729
error 11: 7.47224201129576e-5
order of convergence 2 for n = 11 0.0149616587259972
error 12: 9.35259899040308e-17
order of convergence 2 for n = 12 1.67506033507551e-8
p = 0 after 13 iterations.
```

Based on the results above, the error does decrease over all iterations and converges to 0 (the right answer). For $x_0 = 0.1, 1, 10$ with an order of convergence $\alpha = 2$, it appears that $\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \lambda = 0$, thus it is consistent with my prediction. However, the convergence towards the asymptotic constant is faster and more noticible for $x_0 = 0.1, 1$ than for $x_0 = 10$ as the former decreased after each iteration while the latter sometimes increased.