NUMERICAL METHODS

Quiz 1 Solved Section 81603

ID 441103429
Ibrahim Alrayes

Solved the quiz Using Python Programming Language

1. **Find** the absolute error and error bound for the second approximation of the square root of 19 lying in the interval [4, 4.5] using bisection method. [3 Marks]

```python
[1]
import numpy as np
```

```python
[2]
# Q1
def f(x):
    return x**2-19
def bisection(x0,x1):
    counter = 1
    while counter <= 2:
        x2 = (x0 + x1)/2
        print('Iteration-{}, c{} = {} and f(x2) = {}'.format(counter,counter, x2, f(x2)))
        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2
        counter = counter +1
    abs_error = np.abs(np.sqrt(19) - x2)
    error_bound = ((4.5 - 4) / 2 **2)
    print('\nRoot is : {}\nAbsolute Error is {}\nError Bound is {}'.format(x2, abs_error, error_bound))

bisection(4,4.5)
```

```
Iteration-1, c1 = 4.25 and f(x2) = -0.9375
Iteration-2, c2 = 4.375 and f(x2) = 0.140625

Root is : 4.375
Absolute Error is 0.016101056459326024
Error Bound is 0.125
```

2. **Show** that the Newton's formula for the approximate roots of the quadratic equation $x^2 - k_1 x + k_2 = 0$, where k_1 and k_2 are real numbers, is

$$x_{n+1} = \frac{x_n^2 - k_2}{2x_n - k_1}, \qquad n \geq 0,$$

then use this formula to **find** the second approximation of the positive root of the equation $x^2 - 3x = 4$, starting with $x_0 = 3.5$. [4 Marks]

```python
[20]
from sympy import *
# Q2
x, k1, k2 = symbols('x k1 k2')
expr = x**2 -k1*x +k2
print("Expression : {} ".format(expr))

# Finding the dirivative
expr_diff = Derivative(expr, x)
y = x - (expr/expr_diff)
# Simplifying Newton expression
y = simplify(y)
y = simplify(y) # Again
f = lambdify([x,k1,k2],y)
print("Derivative of expression with respect to x : {}".format(expr_diff))
print("Value of the derivative : {} ".format(expr_diff.doit()))
print("Simplify of Newton Method : {} ".format(y))
print('First approximation {}'.format(f(3.5,3,-4)))
print('Second approximation {}'.format(f(4.0625,3,-4)))
```

```
Expression : -k1*x + k2 + x**2
Derivative of expression with respect to x : Derivative(-k1*x + k2 + x**2, x)
Value of the derivative : -k1 + 2*x
Simplify of Newton Method : (-k2 + x**2)/(-k1 + 2*x)
First approximation 4.0625
Second approximation 4.000762195121951
```

3. **Find** the first approximation of the multiple root of the nonlinear equation $x^3 - 4 = 5x^2 - 8x$ using the best iterative method, starting with $x_0 = 1.75$. [3 Marks]

```
# Q3
def f(x):
    return x**3 - 5*x**2 +8*x -4
def f_first_dir(x):
    return 3*x**2 -10*x +8
def f_second_dir(x):
    return 6*x -10
x0 = 1.75
x1 = x0 - (f(x0) * f_first_dir(x0)) / ((f_first_dir(x0))**2 - (f(x0) * f_second_dir(x0)))
print('First approximation of Second Modified Newton is {}'.format(x1))
```

```
First approximation is 1.9473684210526316
```