Numerical Computing (CS2008)

Sessional-I Exam

Date: Feb 27th 2024

Course Instructor (s)

Dr. Mukhtar Ullah, Dr. Imran Ashraf,
Dr. Muhammad Ali, Muhammad Almas Khan

Total Time: 1 Hour
Total Marks: 55
Total Questions: 03

Semester: SP-2024
Campus: Islamabad
Dept. Computer Science

Solution

________________________ ________ ________ ____________________

Student Name | Roll No | Section | Student Signature

________________________ ____________________

Vetted by | Vetter Signature

Instructions:
Follow these instructions for the source-code:
- Properly indent your code
- Use meaningful names of variable
- For the important parts, use comments to explain your code

Q1: [2+3 +2 + 4 + 4 = 15 marks]

a) Consider the Figure 1 & Figure 2 below.

```
[25]: total = 0.0
      for i in range(1, 10):
          total += 0.1
          print(total)

0.1
0.2
0.30000000000000004
0.4
0.5
0.6
0.7
0.7999999999999999
0.8999999999999999
```

Figure 1

```
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
```

Figure 2

i) Name the type of error in Figure 1 above and mention the source of that error.

Name: **Floating point error.**
Source: **Because of the gap in floating point numbers**

ii) Modify the above code to get the output like mentioned Figure 2.

```
total = 0.0
for i in range(1, 10):
   total += 0.1
   print(round(total,2))
```

b) Consider the following function.

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

The approximate value of above given function cos (3.14) up to the 2nd term is -3.9298 while the value of cos (3.14) approximated through device using 20 terms is: -0.9999987317275388. Now answer the following questions

i) What type of error in the computation of cos (3.14) upto 2nd term and 20th term? Mentions the exact name.

Error type: Discretization error
Source: when we approximate the exact problem by a problem that computers can solve

ii) What is the absolute error between cos (3.14) $_{\text{up to 2nd term}}$ **= -3.9298 and cos (3.14)**$_{\text{up to 20th term}}$**= -0.9999987317275388.**

|-3.9298 – (-0.9999987317275388)| =2.92980126827

iii) What is the relative error between cos (3.14) $_{\text{up to 2nd term}}$ **= -3.9298 and cos(3.14)**$_{\text{up to 20th term}}$**= -0.9999987317275388.**

|-3.9298 – (-0.9999987317275388)| / 3.9298 = 0.7455344466

Q2. [7 + 4 + 2 = 20 marks]

a) **Taylor series are extremely powerful tools for approximating functions that can be difficult to compute otherwise. Find the first 4 terms of the Taylor's series expansion of f(x) = e^x + cos(x) centered around 0**

$$f(x) = e^{x} + \cos(x)$$

$$f(x) = f(0) + f'(0)(x-0) + \frac{f''(0)}{2!}(x-0)^2 + \frac{f'''(0)}{3!}(x-0)^3 + \cdots$$

$$f(0) = e^{0} + \cos(0) = 2$$

$$f'(x) = e^{x} - \sin(x) \Rightarrow f'(0) = 1 - 0 = 1$$

$$f''(x) = e^{x} - \cos(x) \Rightarrow f''(0) = 1 - 1 = 0$$

$$f'''(x) = e^{x} + \cos(x) \Rightarrow f'''(0) = 1 + 1 = 2$$

$$\therefore f(x) = 2 + 1 * x + 0 + \frac{2}{6}x^3 + \cdots$$

$$\Rightarrow f(x) = 2 + x + 0 + \frac{x^3}{3} + \cdots$$

b) **Implement a function in python with the name f(x) which should implement this Taylor's series approximation using these 4 terms**

```python
import numpy as np

def f(x):
    # 4 terms
    term0 = 2
    term1 = x
    term2 = 0
    term3 = 1/3*(x)**3

    # taylor approximatation of y
    y_approx = term0 + term1 + term2 + term3
    return y_approx
```

c) Provide the python code which should call this function f(x) to find the value of f(1.5) and print the result.

```python
x = 1.5
y = np.exp(x) + np.cos(x)
y_approx = f(x)
print('f(1.5) = ', y)
print('f(1.5) approx = ', y_approx)
```

d) What will be order of the error in our approximation in Big-O notation?

$$f^{IV}(x) = e^{x} - \sin(x) \Rightarrow f^{IV}(0) = 1 - 0 = 0$$

Next term $= \dfrac{f^{IV}(0)}{4!}(x-0)^4 = \dfrac{1}{4!}x^4$

$\therefore$ order of error $= O(x^4)$

Given $f(x) = x^3 - x - 1$:
For x=1 f(x) is negative
For x=2 f(x) is positive
Therefore the roots lies between x=1 and x=2 the other values can be seen from the following table.

Iteration no	X_1	X_2	$X_3 = \frac{x_1+x_2}{2}$	$F(x_1)$	$F(x_2)$	$F(x_3)$
1	1	2	1.5	-1	3	0.125
2	1	1.5	1.25	-1	0.125	-0.609375
3	1.25	1.5	1.375	-0.609375	0.125	-0.291015625
4	1.375	1.5	1.4375	-0.291015625	0.125	-0.095947265625
5	1.4375	1.5	1.46875	-0.095947265625	0.125	0.011199951171875
6	1.4375	1.46875	1.453125	-0.095947265625	0.011199951171875	-0.043193817138671875
7	1.453125	1.46875	1.4609375	-0.043193817138671875	0.011199951171875	-0.01620340347290039
8	1.4609375	1.46875	1.46484375	-0.01620340347290039	0.011199951171875	-0.0025535225868225098
9	1.46484375	1.46875	1.466796875	-0.0025535225868225098	0.011199951171875	0.004310242831707001
10	1.46484375	1.466796875	1.4658203125			0.0008751200512051582

b. Solution

Newton's method algorith

1. Set a tolerance TOL for the accuracy
2. Set the maximum number of iteration $MAXIT$
3. Set $k = 0$
4. Initialize x_k and $Error = TOL + 1$
5. while $Error > TOL$ and $k < MAXIT$ do
    - if $f'(x_k) \neq 0$ then
        - Compute $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
        - Set $Error = |x_{k+1} - x_k|$
        - Increase the counter $k = k + 1$
    - end if
6. end while
7. Return the approximation of the root x^*