CIS*3490 A1

Ankush Madharha | 1172859

February 3, 2023

I pledge all work submitted in this document is my own.

Question 1a

Using the formula for error $R_n(x)$ in the Taylor Polynomial approximation, we know that the difference between $f(x) = e^{2x}$ and $P_n(x)$ can be expressed as:

$$e^{2x} - P_n(x) = R_n(x) = \frac{f^{(n+1)}(c)x^{n+1}}{(n+1)!}$$

for some c between 0 and x. Given that $x \in [-1, 1]$, it follows that $c \in (-1, 1)$. To ensure the n is sufficiently large so that $P_n(x)$ is within 10^{-4} of e^{2x}, we aim to select n large enough so that:

$$\left|e^{2x} - P_n(x)\right| = \left|\frac{f^{(n+1)}(c)x^{n+1}}{(n+1)!}\right| = \frac{\left|f^{(n+1)}(c)\right|\left|x\right|^{n+1}}{(n+1)!} < 10^{-4}$$

for all $x \in [-1, 1]$ and $c \in (-1, 1)$. To establish a bound $f^{n+1}(c)$, we remark that the derivatives of $f(x) = e^{2x}$ follow a pattern as followed:

$$f^{(1)}(x) = 2e^{2x}, f^{(2)}(x) = 4e^{2x}, f^{(3)}(x) = 8e^{2x}...$$
$$f^{(n)}(x) = 2^n \cdot e^{2x}$$

Next, we must recognize that for all $x \in [-1, 1]$ we have that:

$$\frac{2^{(n+1)}e^{2c}}{(n+1)!}x^{n+1} < \frac{2^{(n+1)}e^{2(1)}}{(n+1)!}(1)^{n+1}$$

If we set that quantity on the right side to be less than 10^{-4}, so must be the left side of the inequality. We obtain:

$$\frac{2^{(n+1)}e^2}{(n+1)!} < 10^{-4}$$

Now, we can try plugging n values until the inequality is satisfied. We will start at $n = 8$ (a random choice):

$$n = 8 : \ \frac{2^{(9)}e^2}{(9)!} < 10^{-4} \approx 0.01042... \not< 10^{-4}$$
$$n = 9 : \ \frac{2^{(10)}e^2}{(10)!} < 10^{-4} \approx 0.00208... \not< 10^{-4}$$
$$n = 10 : \ \frac{2^{(11)}e^2}{(11)!} < 10^{-4} \approx 0.00037... \not< 10^{-4}$$
$$n = 11 : \ \frac{2^{(12)}e^2}{(12)!} < 10^{-4} \approx 0.00006... < 10^{-4}$$

$n = 11$ is the first value for which this inequality is true. That means we would need a 11^{th} order Taylor Polynomial approximation to ensure that it is accurate to within 10¯4 for all $x \in [-1, 1]$.

Question 1b

```
% Course: MATH*2130
% Name: Ankush Madharha
% Student Number: 1172859

% Prompt user for parameter inpt and error threshold input
k = input("Enter parameter k: ");
err = input("Enter error threshold: ");

% Define our error function
n=0;
f = @(n) (k.^(n+1)*exp(1).^(k)) / (factorial(n+1));

% Continously loop while incrementing n error function < error
while f(n) >= err
    n = n + 1;
end

disp("n = " + n);
```

Sample Output

```
>> Question_1b
Enter parameter k: 2
Enter error threshold : 10^-4
n = 11
```

Question 2a

$$\lim_{n\to\infty} \frac{2e^{2n} - e^n + \ln(n) + n^2}{e^{2n}}$$

$$= \lim_{n\to\infty} \frac{2e^{2n}}{e^{2n}} - \frac{e^n}{e^{2n}} + \frac{\ln(n)}{e^{2n}} + \frac{n^2}{e^{2n}}$$

We will solve these 4 limits individually:

$$\lim_{n\to\infty} \frac{2e^{2n}}{e^{2n}} = 2 \cdot \lim_{n\to\infty} \frac{e^{2n}}{e^{2n}} = 2 \cdot \lim_{n\to\infty} 1 = 2 \cdot 1 = 2$$

$$\lim_{n\to\infty} \frac{e^n}{e^{2n}} = \lim_{n\to\infty} e^{n-2n} = \lim_{n\to\infty} e^{-n} = \lim_{n\to\infty} \frac{1}{e^n} = \text{"}\frac{1}{\infty}\text{"} = 0$$

$$\lim_{n\to\infty} \frac{\ln(n)}{e^{2n}} = \lim_{n\to\infty} \frac{1/x}{2e^{2n}} = \lim_{n\to\infty} \frac{1}{x \cdot 2e^{2n}} = \text{"}\frac{1}{\infty}\text{"} = 0$$

$$\lim_{n\to\infty} \frac{n^2}{e^{2n}} = \lim_{n\to\infty} \frac{2n}{2 \cdot e^{2n}} = \lim_{n\to\infty} \frac{1}{2 \cdot e^{2n}} = \text{"}\frac{1}{\infty}\text{"} = 0$$

Now, plugging these limit results into our original:

$$2 - 0 + 0 + 0 = 2$$

Since our limit results in a finite constant, this proves that $h(n) = 2e^{2n} - e^n + \ln(n) + n^2 \in \mathcal{O}(e^{2n})$

Question 2b

```
% Course: MATH*2130
% Name: Ankush Madharha
% Student Number: 1172859

% Prompt user for function input, L input, and error threshold
f = input('Enter a function: ', 's');
L = input('Enter a limit L: ');
err = input('Enter your error threshold: ');

% Convert string function to a function handle of n
n = 1;
f = "@(n)"+f;
f = str2func(f);

% Loop until abs(f(n) - L) is greater than or equal to threshold
while abs(f(n) - L) >= err
    n = n + 1;
end

disp("n = " + n);
```

Sample Output

```
>> Question_2b
Enter a function: (2*exp(1)^(2*n) - exp(1)^(n) + log(n) + n^2) / (
    exp(1)^(2*n))
Enter a limit L: 2
Enter your error threshold: 10^-6
n = 14
```

If you copy and paste this code into MATLAB, it puts a space after the 's' in the function input statement (which isn't there in the original file), resulting in an error. Besides that, it should work as expected.