Homework 3 Report

Group 334-7

Group Work Overview:

HW2	**Cameron Rabiyan**	**Maya Apotheker**	**Manny Gamboa**
Coding	**20.0%**	**40.0%**	**40.0%**
Results	**0%**	**50.0%**	**50.0%**
Report	**0%**	**50.0%**	**50.0%**
Overall		**Driver/Navigator**	**Driver/Navigator**

Introduction Part 1:

Using Fortran, we aim to create a program that can approximate pi using summations while comparing it to pi calculated using $acos(-1.d0)$.

Procedures: Pi

1. Write Code that Approximates π_{approx}, using the following summation:

$$\sum_{n=0}^{\infty} 16^{-n}\left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6}\right)$$

2. We calculated the difference between π_{approx} and π_{true}, calculated using $\pi_{true} = acos(-1.d0)$.

   $\pi_{approx} - \pi_{true} = \pi_{diff}$, where π_{diff} is greater than our threshold.

3. Calculate the number of terms summed (N) and the π_{diff} for four different thresholds: 1.e-4, 1.e-8, 1.e-12, and 1.e-16
4. Used makefile with corresponding extensions to compile the code.

Results for Part 1:

Threshold	**Pi Approximate**
1e-4	3.34222
1e-8	3.34222
1e-12	3.34222
1e-16	3.34222

Part 1 Summary:

As the threshold was decreased, the absolute value difference between our π_{approx} and π_{true} increased, this allowed for more terms within our summation to be summed resulting in a closer approximate to our π_{true}. Unfortunately, we were not able to get the proper pi approximate values which should've improved with smaller thresholds. Using the programs created, we made a makefile which allows for the text documents used to become executables without the aid of gfortran compilation.

Introduction: Part 2

Using FORTRAN, we aim to approximate integration using the trapezoidal approximation rule of calculus.

Procedures: Trapezoidal Rule

1. Write code that allows us to represent:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + ... + 2f(x_{n-1}) + f(x_n)] \text{, where } \Delta x = \frac{b-a}{N} \text{ and } x_i = a + i\Delta x$$

2. Using this code, we will test the function $f(x) = \int_0^1 x^2 + 1dx$
3. Test the following values of n and see how the affects the answer: 25, 50, 100, 200, and 400.

Part 2 Results:

n	**Approximation for (**$\int_0^1 x^2 + 1dx$**)**
n=25	1.35354601
n=50	1.3439201
n=100	1.33834898
n=200	1.33583736
n=400	1.33458436

Part 2 Error:

Output:	**Error within Function:** (TrapezoidExact - TrapezoidFun)	**Error within Subroutine:** (TrapezoidExact - TrapezoidSub)

1.35354601		
1.3439201	-2.02062e-002	-2.02062e-002
1.33834898		
1.33583736		
1.33458436		

Part 2 Conclusion:

As listed in the results, the larger the value of n, the closer our approximation of the integration $x^2 + 1$ where x is between [0,1]. The Trapezoid Rule states that the larger the value of n, the smaller the ΔX, the smaller the increments that result in smaller, but more trapezoids. These trapezoids are then summed to result in the approximation used in the Trapezoidal Approximation rule of calculus. Our expected result for the output, using both pen and paper and the many calculus calculators available on the internet for verification, was exactly $\left(\frac{4}{3}\right)$ or $1.\overline{33333}$. As observed in our results, the larger the n, or the more trapezoids that we allow to be summed, the closer to $1.\overline{33333}$ we receive from our program.