**MAE 215 MATLAB Final Project Report**

Bo-Chang, Lin

Ira A. Fulton Schools of Engineering

Arizona State University

MAE 215: Intro to Programming in MATLAB

Dr. Joshua Wilbur

April 28, 2023

**Table of contents**

1. **Introduction** Page 3
2. **Task 1 - Generate Taylor Series with User-defined Function: TaylorSeriesFUN**

i.) Problem Statement Page 4

ii.) MATLAB Script Page 5

iii.) The Outputs Page 6

iv.) User-defined Function – TaylorSeriesFUN Page 7

1. **Task 2 - Calculates Center of Mass for Particles with User-defined Function: MassCenterSUM**

i.) Problem Statement Page 8

ii.) MATLAB Script Page 9

iii.) The Outputs Page 10

iv.) User-defined Function – MassCenterSUM Page 11

v.) Input Data - MassData1.xlsx Page 13

**Introduction**

In the course of MAE 215: Introduction to Programming in MATLAB, we have learned several essential coding concepts including conditional statements, looping structures, functions, and symbolic math. The objective of this course is helping us as engineering students to be prepared for entering both academic and career path. All engineering students have to be familiar with at least one programming language. Taking MATLAB for example, it is a powerful tool for engineering computation and data visualizing. Engineers are responsible for making decisions, and computer can do all the repetitive tasks efficiently for us as long as we have enough knowledge on programming.

**Generate Taylor Series with User-defined Function: TaylorSeriesFUN**

**Problem statement**

In mathematics, Taylor Series is a useful tool for approximating a value at a point on a math function. Finding a Taylor Series of any math function manually can be a daunting task. Therefore, I created a MATLAB function that generates Taylor Series.

Taylor Series formula:

*f(x) = f(a) + f’(a)(x-a) + (f’’(a)/2!)(x-a)2 + … +(fn(a)/n!)(x-a)n*

To run the function – TaylorSeriesFUN, here is the list of required input data.

1. **f**: any original math function needs to find its Taylor Series representation. (Symbolic expression)

2. **aValue**: the point the math function is based on (Data type: double)

3. **n**: number of terms you want to find (Data type: double)

4. **xValue**: a point for test out y value (Data type: double)

For the output:

1. **C**: List of the coefficients of the Taylor Series in a row matrix (Data type: double)

2. **TaylorEXP**: the new math function in Taylor Series representation (function handle)

3. **Err**: the aboslute error between actual value and approximated value. (Data type: double)

Aside from the function, the script in the end also provides the code for plotting the two functions.

**Calculates Center of Mass for Particles with User-defined Function: MassCenterSUM**

**Problem Statement**

In physics, finding the center of mass of an object is crucial to calculate many other physical properties. This function is designed to solve simplified tasks – finding the center of mass for particles system. The function MassCenterSUM has flexible number of input varables ranging from 2 to 4, but it has to be in the order of **xi**, **yi**, **zi**, and **mi,** representing x, y, z coordinates and individual mass. Each of them is a single column matrix imported from external data files. For the output, the function simply returns the corresponding coordinates of its center of mass. To be clear, plug in **xi**, **yi** then **mi** as 3 inputs for getting (mx, my). Also the function provides visualize data shows masses distribution and marks the center of mass in red.

The formula for calculating center of mass:

COM(x) = SUM(xi\*mi)/M; COM(y) = SUM(yi\*mi)/M; COM(z) = SUM(zi\*mi)/M