Project 6: Numeric Computations with Taylor Polynomials

Austin Johns and Shelby Clow

Grand Canyon University

CST-305: Principles of Modeling and Simulation

**Required Software and Libraries**

In this project, numpy, sympy, odeint and pyplot are all used to calculate and visualize the results of the Taylor series.

**A picture containing bottle, sitting, holding, people

Description automatically generated**

**Program Execution**

The execution of this program includes beginning with importing the necessary packages before defining the Taylor function.

A screenshot of a cell phone

Description automatically generated

Afterwards, the function for the factorial calculation is made to be used in expansion.

A picture containing drawing

Description automatically generated

Then, the results of the Taylor Series are stored in an array for later use. Here, the range function is used to return multiple results, as the Taylor series would for a1, a2, …, an.

A picture containing drawing

Description automatically generated

After that, the differential equation is plugged into the Taylor function, returning the results using the equation.

A picture containing clock

Description automatically generated

Lastly, output of the results with reference to the array.

A close up of a keyboard

Description automatically generated

**Mathematical Methods in Code**

The mathematical methods in the code are as follows:

First, the recurrence formula is used to the degree of the equation. Then, the Taylor Series function (as seen above) uses ], for example, to define different degrees of derivatives. Next, the result of the recursion is referenced in the Taylor function to get the output of the series. Lastly, the program outputs the various results of the recursion in the code.

**Program Output**

The output of the program states the n value, which is 4 for part 1 and 8 for part 2.

Afterwards, the program states the value around the point given. Below this value, the program outputs the approximation of the Taylor Series given n around the point 3.5. After this, the same process is repeated for part 2.







**Programming Style**

The programming style for this project is fairly straightforward. All code is commented and organized with clear order and structure. This is evident as each step of the Taylor Series can be seen separated by the different processes, carried over into the code.

**Mathematical Methods**

**A document with black text

Description automatically generatedA screenshot of text

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA close up of a piece of paper

Description automatically generated**

**Execution Screenshots**

**A close up of text on a black background

Description automatically generated**

A screenshot of a computer

Description automatically generated

**README**

**A screenshot of a cell phone

Description automatically generated**