Taller de Desarrollo de Aplicaciones

Práctica 1. Predicción de respuesta dinámica de sistemas lineales e invariables en tiempo (LTI)
Fechas de entrega:
- 13 de Febrero, 11:59 horas: Entrega de la especificación, ver sección de Desarrollo
- 20 de Febrero, 11:59 horas: Entrega de diseño del software y primera entrega (código parcial).
- 27 de Febrero, 11:59 horas: Entrega de codificación, documentación y demostración de sistema.

Introducción

Varios de los procesos en ingeniería y ciencias pueden ser modelados a través de ecuaciones diferenciales lineales y de parámetros constantes. Sin embargo, en la naturaleza, la mayoría de los procesos no son realmente caóticos o no-lineales. Estos sistemas no pueden ser analizados por métodos analíticos tradicionales (sustituciones, separabilidad, transformaciones), por lo que el análisis y predicción de su comportamiento estaba fuera de nuestro alcance. Gracias a avances en sistemas computacionales, han surgido alternativas para el análisis de comportamientos de sistemas complejos con el fin de intentar predecir su comportamiento.

Un método de solución de estos modelos matemáticos es a través de soluciones numéricas. Gracias a estos podemos estudiar estos sistemas de manera objetiva con ciertas restricciones. De los métodos mas usados, se encuentra el método desarrollado por Euler, denominado como método de Euler, el cual nos permite predecir el comportamiento de una function cuyos valores iniciales son conocidos (1).

$$\frac{dy}{dt} = f(t, y) \quad y(t_0) = y_0 \quad (1)$$

Donde $f(t, y)$ es una función conocida y las condiciones iniciales, $y(t_0) = y_0$, son conocidos. Si la función es continua y finita en domino, entonces hay una solución única para el intervalo $t_0 < t < t_1$. Esto nos permite, mediante un método iterativo, aproximar la solución de la ecuación mediante la información inicial obtenida.

La información inicial es el valor de la solución en tiempo $t = t_0$. Su derivada en tiempo, por definición, tiene que tocar a la función en el mismo punto (2).

$$\frac{dy}{dt}\Big|_{t=t_0} = f(t_0, y_0) \quad (2)$$

Si interpolamos esta linea en tiempo por un incremento $t_1 - t_0 = h$, entonces podemos calcular la recta tangente a la función y aproximar el siguiente valor de, y, en tiempo (3) Figura 1:

$$y_1 = y_0 + f(t_0, y_0)(t_1 - t_0) \qquad (3)$$

Si de manera recursiva, implementamos este proceso, podemos lograr una aproximación total a la función $y(t)$.
Un algoritmo para realizar esta actividad puede ser desarrollado utilizando arreglos lineales por su naturaleza secuencial en tiempo.

Objetivos

- Definir y escribir la especificación de la práctica: algoritmo general, pseudocodigo de funciones, estructura del algoritmo, literatura básica, y aproximación de complejidad en espacio y tiempo.
- Diseñar e implementar el código que permita:
    - Definir una ecuación diferencial de orden $y^{(m)}$ dada por el usuario, y coeficientes $a_0,\ a_1\ \ \ldots a_{n-1}$, mediante un arreglo estático.
    - Resolver la ecuación diferencial por medio del método de Euler a una función de entrada senoidal o una constante, definida por el usuario.
    - Entradas: $t_0, t_f, y_0, h\ \ y\ \ n$.
    - Imprimir los resultados en un file, "solucion.dat", cuyo formato debe ser de texto y ordenado de la siguiente manera: la primer columna corresponderá al valor de tiempo y la segunda al valor de la predicción.
    - Plot de la señal de salida mediante GNU Plot.

Alcances y Limitaciones

- Es indispensable utilizar buenas prácticas de codificación y documentación
- Se debe programar modularmente.
- En una primer parte, el grado de la ecuación diferencial m, sera máximo de orden 3, sin embargo, el código debe ser independiente a la estructura de dato utilizado para generar la ecuación diferencial.
- Se deben usar archivos de texto.
- Se deben dar argumentos desde la línea de comandos.
- Se deben mostrar elementos para la prueba y verificación del código mediante ecuaciones diferenciales ordinarias. eg. : $y' + 2y = 2 - e^{-4t} \qquad y(0) = 1.$
- Se debe validar contra errores de entrada del usuario mediante códigos de error: enums().

Metodología

a) Reunirse en equipos de 2. Realizar el pseudocodigo, documentar la literatura básica y su análisis de complejidad en tiempo y espacio inicial.

b) **Entregable:**
(13 de Febrero, 11:59 horas): Especificaciones por equipo, con el nombre de los integrantes y la especificación final, también con el nombre de los que participaron.

c) De acuerdo a la especificación se deberá analizar, diseñar e implementar el código.
Entregables:
(20 de Febrero, 11:59 horas): Diseño e implementación del solucionador de ecuaciones diferenciales con listas lineales, software de escritura en archivo de texto y script de GNU plot.
(27 de Febrero, 11:59 horas): Implementación del solucionador de ecuaciones diferenciales con listas lineales, software de escritura en archivo de texto y script de GNU plot junto con documentación de operación y muestras.

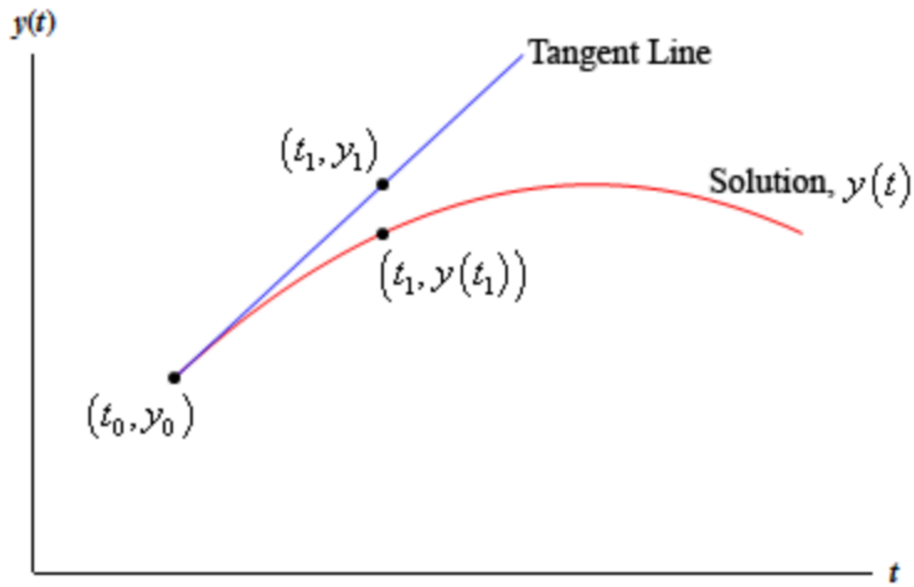

Figura 1. Aproximación de funciones por el método de Euler.

Nota: Codificar el solucionador para ecuaciones diferenciales de orden m dado por el usuario, amerita 3 puntos extras sobre la práctica (Hint: Arrays dinámicos).