Práctica 1

Predicción de respuesta dinámica de sistemas lineales e invariables en tiempo (LTI)

Integrantes del equipo

César Mauricio Arellano Velásquez

Raúl González Portillo

Profesor

César Ángeles

Materia

Taller de Desarrollo de Aplicaciones

Introducción:

En [matemática](https://es.wikipedia.org/wiki/Matem%C3%A1tica) y [computación](https://es.wikipedia.org/wiki/Computaci%C3%B3n), el **método de Euler**, llamado así en honor a [Leonhard Euler](https://es.wikipedia.org/wiki/Leonhard_Euler), es un procedimiento de [integración numérica](https://es.wikipedia.org/wiki/Integraci%C3%B3n_num%C3%A9rica) para resolver [ecuaciones diferenciales ordinarias](https://es.wikipedia.org/wiki/Ecuaciones_diferenciales_ordinarias) (EDO) a partir de un valor inicial dado. El método de Euler es el más simple de los [métodos numéricos](https://es.wikipedia.org/wiki/M%C3%A9todos_num%C3%A9ricos) para resolver un [problema de valor inicial](https://es.wikipedia.org/wiki/Problema_de_valor_inicial), y el más simple de los [Métodos de Runge-Kutta](https://es.wikipedia.org/wiki/M%C3%A9todo_de_Runge-Kutta). El método de Euler es nombrado por [Leonhard Euler](https://es.wikipedia.org/wiki/Leonhard_Euler), quien lo trató en su libro *Institutionum calculi integralis* (publicado en 1768-1770).

El método de Euler es un método de primer orden, lo que significa que el error local es proporcional al cuadrado del tamaño del paso, y el error global es proporcional al tamaño del paso. El método de Euler regularmente sirve como base para construir métodos más complejos.

Consiste en dividir los intervalos que va de x0 a xf en n subintervalos de ancho h; o sea:



de manera que se obtiene un conjunto discreto de n + 1 puntos: x0, x1, x2, …, xn del intervalo de interés [x0, xf]. Para cualquiera de estos puntos se cumple que:

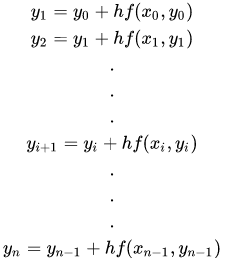


La condición inicial , representa el punto por donde pasa la curva solución de la ecuación del planteamiento inicial, la cual se denotará como Ya teniendo el punto se puede evaluar la primera derivada de en ese punto; por lo tanto:

Se resuelve para 



Es evidente que la ordenadacalculada de esta manera no es igual apues existe un pequeño error. Sin embargo, el valor sirve para que se aproxime en el punto y repetir el procedimiento anterior a fin de generar la sucesión de aproximaciones siguiente:



Objetivos:

* Comprender de manera básica el funcionamiento de algoritmos de predicción a través de métodos matemáticos para tener una mejor toma de decisiones.
* Determinar una correcta complejidad de tiempo y espacio para optimizar procesos y codificación.
* Mejorar habilidades de investigación para resolución de problemas.
* Comprender el método de Euler para obtener la curva solución para predecir el comportamiento de una función.

Análisis

Pseudocódigo:

**#define return = retorna el valor de alguna operación o variable.**

**#define fopen = Abre el archivo (de texto o binario) que se le indica y junto al modo de ejecución.**

**#define fclose = Cierra el archivo que se le indique y que previamente esté abierto.**

**#define fprintf = Imprime en el archivo especificado.**

**#define wt = Abre el archivo en modo de escritura.**

Principal ( | )

{

PedirDatos ( | Y0, T0, H, Tf);

Diff\_Solver (Y\_Array, T\_Array, Y0, T0, H, Tf | Limite);

ImprimirArch (Y\_Array, T\_Array, Limite | );

}

PedirDatos ( | Y0, T0, H, Tf)

{

Imprimir (“Introduzca los siguientes datos:”);

Imprimir (“T0:”);

Leer (T0);

Imprimir (“Y(T0)”);

Leer (Y0);

Imprimir (“H”);

Leer (H);

Imprimir (“Tf”);

Leer (Tf);

}

Funcion (Y0, T0 | )

{

return T0+Y0; //Función matemática establecida (x+y).

}

Diff\_Solver (Y\_Array, T\_Array, Y0, T0, H, Tf | Limite)

{

Desde i = 0; Hasta T0 <=Tf; i = i + 1

{

Y\_Array[i] = Y0;

T\_Array[i] = T0;

Y0 = Y0 + H \* (Funcion(T0,Y0));

T0 = H + T0;

}

↑Limite = i;

}

ImprimirArch (Y\_Array, T\_Array, Limite | )

{

Archivo = fopen(“LTI.txt”,”wt”);

desde i=0; hasta i < Limite; i = i + 1

{

fprintf (Archivo, T\_Array[i],” ”, Y\_Array[i]);

}

fclose(Archivo);

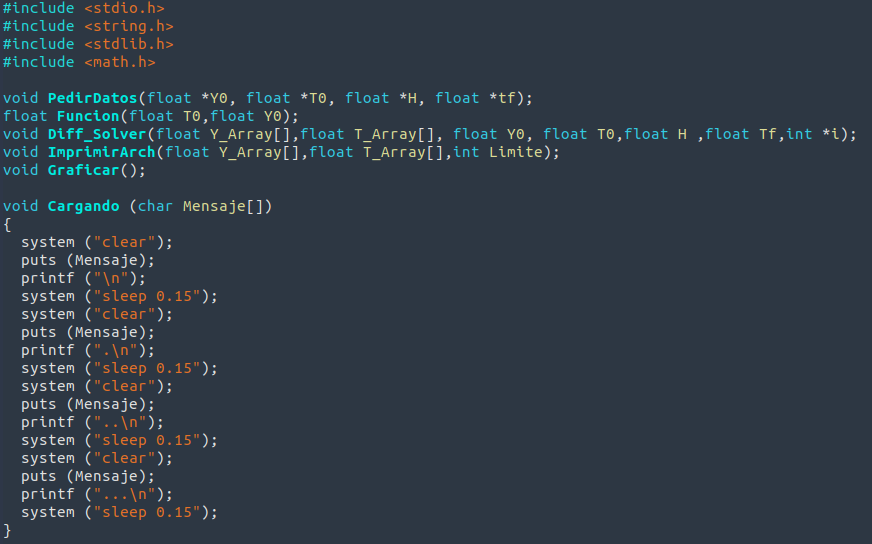
}

**Diagrama IPO**



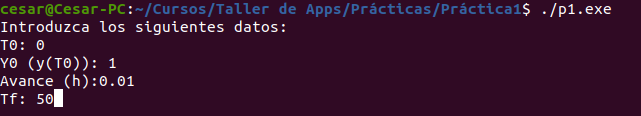
**Entradas Procesos y Salidas**

|  |  |
| --- | --- |
| Entradas | |
| **Nombre** | **Descripción** |
| Y0 | Guarda la evaluación de la función Y en T0 (Y(T0)) |
| T0 | Guarda el valor inicial T0 |
| h | Guarda el valor del step que dará la función por cada iteración |
| Tf | Indica el Tiempo donde terminará la gráfica |
| Procesos | |
| **Nombre** | **Descripción** |
| PedirDatos | Solicita los datos necesarios para realizar la predicción |
| Diff\_Solver | Realiza una predicción en base a los datos del usuario utilizando el método de Euler y la guarda en dos arreglos. |
| ImprimirArch | Imprime los datos de los arreglos Y\_Array[] y T\_Array[] en un formato interpretable por GNUPlot |
| Graficar | Inicializa GNUPlot con los títulos de la gráfica y los ejes y le envía el archivo creado para graficarlo. |
| Salidas | |
| **Nombre** | **Descripción** |
| Limite | Guarda cuantas iteraciones tuvo la función, sirve para determinar hasta donde se debe graficar |
| Y\_Array[] | Un arreglo que guarda los resultados a graficar en el eje Y |
| T\_Array[] | Un arreglo que guarda los resultados a gráficar en el eje X (T) |

**Código**



**Ejecución del programa**





**Demostración a través de gráfica.**

