Taller de Desarrollo de Aplicaciones

Práctica No.1

Método predictivo en tiempo

Giovanni Josué Venegas Ramírez
Castro Bouquet Ildefonso

Abstract. - Mediante el método de Euler de inferencia de sistemas en tiempo se realizará la presente practica donde se aplicará el método de predicción de Euler que ayuda a conocer tiempos futuros a partir de un estado actual, así como las fuerzas que ejercen en dicho contexto a analizar de un individuo en un Bunge jump.

I. Introducción. –

Hoy en día, los nuevos métodos de inteligencia artificial nos permiten encontrar relaciones y patrones incomprensibles y nos ayudan a comprender el comportamiento de muchos fenómenos complejos. Una de estas aplicaciones es un sistema que intenta predecir el estado de los fenómenos durante un período de tiempo. A partir de la sugerencia de comportamiento, expresada matemáticamente, el valor inicial se define por el estado inicial y los valores pasados.

Como seres humanos nos podemos preguntar qué es lo podría suceder si se realizan ciertas acciones, y como tal no hay una manera de saber concretamente el efecto a lo realizado, por lo que se hacen propuestas para poder hacer un estimado a lo que se cree que podría ser el resultado, uno de estos métodos nos permite observar el patrón de comportamiento para poder generar un valor de predicción de nuestro interés.

Este tipo de sistemas siguen evolucionando, de diferentes maneras y diferentes usos, todo a partir de un historial para generar posibles soluciones.

En matemática y computación, el método de Euler, llamado así en honor a Leonhard Euler, es un procedimiento de integración numérica para resolver ecuaciones diferenciales ordinarias (EDO) a partir de un valor inicial dado. El método de Euler es el más simple de los métodos numéricos para resolver un problema de valor inicial, y el más simple de los Métodos de Runge-Kutta.

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. También sirve como base para construir métodos más complejos.

La idea es que a pesar de que la curva es desconocida en un principio, su punto de comienzo, al cual denotamos por A0, es conocido. Entonces, de la ecuación diferencial se puede calcular la pendiente de la curva en el punto A0 y por lo tanto la recta tangente a la curva.

Ahora, dando un pequeño paso sobre dicha recta, podemos tomarnos un nuevo punto A1 y suponer que dicho punto pertenece a la curva, entonces seguimos el mismo razonamiento aplicado anteriormente y volvemos a calcular la pendiente de la recta tangente a la curva en el punto A1. Luego de varios pasos tendremos formada una curva poligonal A0A1A2A3... En general esta curva que obtenemos al aplicar el método no diverge lejos de la curva original, además el error entre ambas curvas se puede minimizar si se dan pasos muy pequeños al avanzar sobre la recta tangente a la curva y además el intervalo sobre el que trabajamos es finito.

II. Contexto

Como parte de nuestro programa, debemos obtener el comportamiento de una persona expulsada de un Bungee para definir instantáneamente su trayectoria después del inicio. El método de Euler es aplicado en la práctica para predecir el comportamiento temporal de la ecuación en obtenida en la práctica como modelo matemático a seguir. Por lo tanto, por las condiciones Iniciales, podemos acercarnos a todos los intervalos de tiempo que se ejecutan en cada iteración teniendo en cuenta que la seguridad de la persona se basa en estos cálculos matemáticos.

III. Análisis

Se diseñará un software a través de modularidad y modelo MVC que prediga un movimiento con trayectoria de un bongee, junto a la información generada se hará una gráfica con GNUplot.

Para ello se utilizarían mínimo tres módulos, uno donde se encuentra el software principal (MOTOR), se usará un ".h" para saber qué procesos se harán mediante qué funciones, y por último un ".c" donde se desarrollen las funciones a utilizar.

Primero hay que analizar nuestro problema, necesitamos variables de entrada, en este caso variables para sustituir en la ecuación de Euler:

→ **ΔT,** Es el tiempo discreto
→ **Masa (M),** La masa de una persona en kg
→ **K,** constante del bongee jump
→ **g,** la constante de gravedad: 9.81 m/s^2
→ Condiciones iniciales

También necesitamos saber qué valores de salida daremos:

→ Se estrella o no contra el suelo
→ Valores para gráfica GNUplot
→ Archivo CSV

Ya con entradas y salidas podemos saber el proceso a realizar, en este caso:

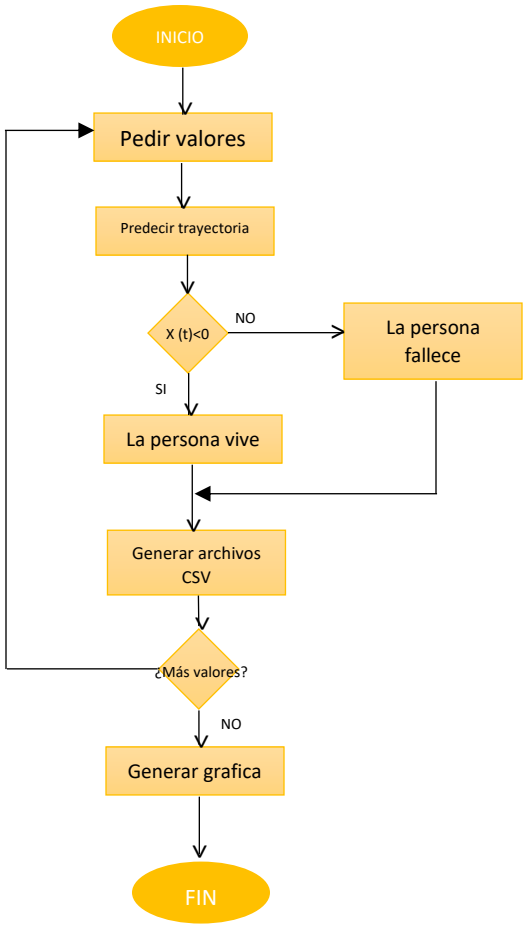
$$x(t+2\Delta t) = -x(t)\left[1+\frac{k\Delta t^2}{m}\right] + 2x(t+\Delta t) + \Delta t^2 g$$

IV. Diseño

El presente desarrollo se basa en el modelo MVC es decir en el modelo-vista-controlador el cual es un estilo de arquitectura de software que separa los datos de aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintitos.

→ **Modelo:** Enfocado a la predicción de la trayectoria.
→ **Vista:** Despliega un menú con las instrucciones del programa para así poder asignar valores
→ **Controlador:** Comparación de posiciones.

Diagrama de flujo de datos (DFD)



V. Pseudocódigo

Se concibe fragmentar el código en módulos que permitan tener una organización optima.

Para este mismo se desarrollarán 3 módulos.

Un maestro, otro para enviar la información a los archivos y un último para el proceso y desarrollo de las ecuaciones.

El módulo principal será la vista, donde colocara un menú, los usuarios podrán insertar el valor(es) de la ecuación. Es decir, el individuo, la constante de elasticidad del Bungee Jump, el intervalo del tiempo (▲t) y la duración total del juego.

El módulo ***files.c*** se utilizará para ingresar la información recogida por el modulo ***equation.c*** para los archivos, este enviará los datos de los valores de la posición y de los datos de tiempo de ejecución del programa.

El módulo ***equation.c*** se utilizará como raíz maestra principal donde se harán los cálculos de la ecuación predictiva de Euler para obtener los datos de la posición de la persona en el intervalo que el usuario ingrese.

main.c

```
void main{
   FILE *archivo1, archivo2;
   int time = user input
   int k= user input
   int masa=user input
   int contador //Contador de ciclos
        clock_t start,stop;
        double cpu_time, resultado;
                archivo1= fopen
("TiempoCPU.CSV", "w")

                archivo2= fopen
("tiempoRES.CSV", "w")
for(contador = 0 ; contador <= time ; contador ++)
                {
   start = now ()
resultado = equation (K, masa, time, contador)
                stop= now();
                FILES
        cpu tiempo - stop-start;

imprimir (cpu_tiempo, contador, resultado)
        resultado -0
                }
        fclose (archivo1)
        flose (archivo2)
}
```

equation.c

```
equation (tiempo, k, masa, contador){
   double res, eq1, eq2, eq3
   m=masa, ▲t=tiempo, t=contador, g=9.81 m/s^2

   eq1= -x(t)[(k▲t^2/m)+1];
   eq2= 2x(t+▲t);
   eq3= eq1+eq2+eq3;
        return res;
}
```

files.c

```
impimir (cpu_tiempo, contador, resultado, archivo1,
archivo2){
fprintf (archivo1, contador/t  resultado /n)
fprintf (archivo2, contador/t  cpu_tiempo /n)
}
```