

# Práctica 1

---



## Método Predictivo en tiempo

### Integrantes del equipo

Giovanni Josue Venegas Ramirez  
Castro Bouquet Ildfonso

### Profesor

César Arturo Angeles Ruiz

### Materia

Taller de Desarrollo de Aplicaciones

**Entregable:** (24 de Septiembre, 11:59 horas): Especificaciones por equipo, con el nombre de los integrantes y la especificación final, también con el nombre de los que participaron.

---

## Introducción:

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  $A_0$ , es conocido. Entonces, de la ecuación diferencial se puede calcular la pendiente de la curva en el punto  $A_0$  y por lo tanto la recta tangente a la curva.

Ahora, dando un pequeño paso sobre dicha recta, podemos tomarnos un nuevo punto  $A_1$  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  $A_1$ . Luego de varios pasos tendremos formada una curva poligonal  $A_0A_1A_2A_3...$ . 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

Se diseñará un software a través de modularidad y modelo MVC que prediga un movimiento con trayectoria de un bungee, 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**:

- $\Delta T$ , Es el tiempo discreto
- **Masa (M)**, La masa de una persona en kg
- **K**, constante del bungee jump
- **g**, la constante de gravedad: 9.81 m/s<sup>2</sup>

También necesitamos saber qué valores de salida daremos:

- Se estrella o no contra el suelo
- Valores para gráfica GNUplot

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

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

El programa será llamado a partir del comando “./bungee”. Este mismo generará a partir de los datos obtenidos un archivo de texto “valBongee.txt” (*Si se desea generar un archivo con nombre propio se introducirá al momento de ingresar los valores de las variables”, para después ser leídos mediante GNUplot*)

Se utilizarán menús para la entrada de datos, si en alguna variable no se recibe valor, se establecerá un valor predeterminado para cada una.

Se creará una Variable de tipo estructura -DATOS-, con las siguientes variables:

- Float tiempo
- Float masa
- Float K

El código utilizará funciones para acciones específicas:

- **Menú**, Parámetros (apuntador a una variable de tipo estructura-DATOS-). Generará un menú para introducir valores y los guardará en la estructura recibida.
- **DesarrolloEcuacion**, Parámetros (apuntador a una variable de tipo estructura-DATOS-, Array Bidimensional para almacenar los valores obtenidos de la ecuación)
- **CreateFile**, Sin parámetros, se encargará de crear una variable de tipo FILE con la dirección del archivo creado. Regresará la variable.
- **FillFile**, Parámetros (Variable con la dirección del archivo creado, Array bidimensional con los datos a copiar al archivo), se encargará de hacer "Write Text" en el archivo recibido con los datos del array.