

Taller de Desarrollo de Aplicaciones

Práctica No. 1

Ulises Magaña Estrada
Isaac Eduardo Odriozola Díaz
José Antonio Pérez Durán

Abstract- Por medio del método de inferencia de sistemas en tiempo se lleva a cabo la siguiente práctica donde se aplica el uso del Método de Predicción de Euler que ayuda a conocer tiempos futuros a partir de un estado actual y las fuerzas que actúan en el fenómeno a estudiar de una persona en un bungee que es jalada desde abajo con una cuerda.

I. Introducción

Hoy en día, nuevas metodologías en inteligencia artificial, nos ha permitido encontrar relaciones y patrones que son difíciles de ver y nos han ayudado a conocer el comportamiento de muchos fenómenos complejos.

Una de estas aplicaciones son los sistemas que intentan predecir el estado de un fenómeno en el tiempo. A partir de una propuesta de comportamiento, expresado de una manera matemática, los primeros valores son definidos por el estado inicial y valores pasados. Una vez que los primeros datos son verificados, estos se retroalimentan dentro del sistema para poder generar objetos no vistos previamente por el usuario que pueden ser de interés.

Estos sistemas no solo son capaces de sugerir nuevos comportamientos acorde al historial del sistema en el pasado; si no también, tienen el gran poder de facilitar el proceso de búsqueda en problemas más complejos como rastreo o generación de rutas. Es por eso por lo que su utilización ha sido considerada de gran importancia en la tecnología actual.

II. Contexto

En el contexto de nuestra aplicación debemos obtener el comportamiento de una persona lanzada desde un Bungee para definir su trayectoria en un instante posterior a su lanzamiento. El método de Euler es aplicado en la práctica para predecir el comportamiento temporal de la ecuación dada en la práctica como modelo matemático.

Es por esto que por medio de las condiciones iniciales podemos realizar una aproximación de todos los intervalos de tiempo que se van realizando en cada iteración considerando que la seguridad del individuo se basa en estos cálculos matemáticos.

III. Análisis

A. Descripción

Se desea desarrollar un programa en C utilizando el método de Euler que predice la trayectoria de una persona al ser jalada con una cuerda desde un bungee.

B. Entradas

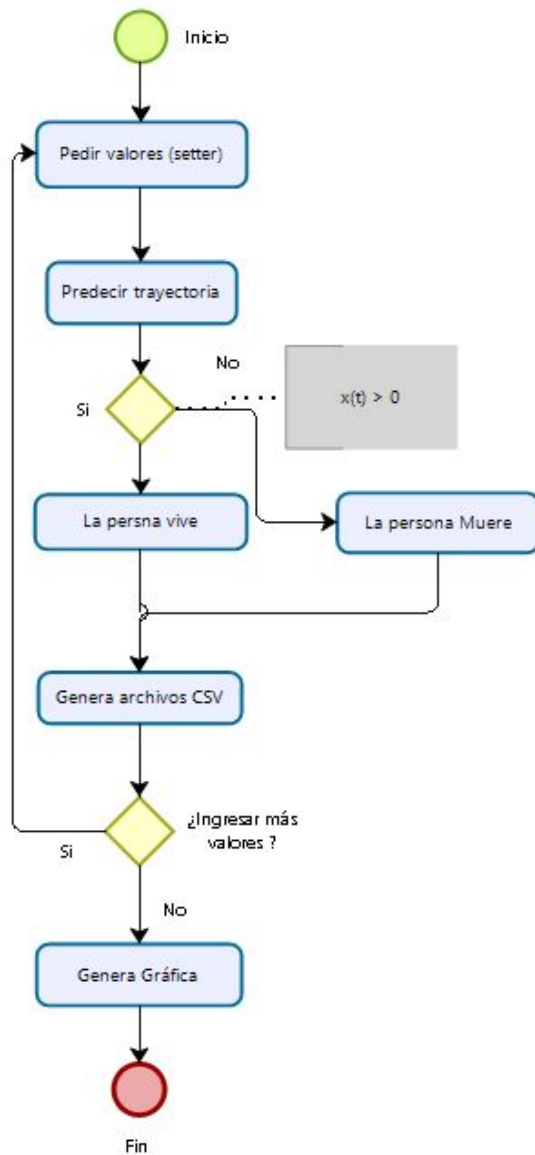
- Masa
- Constante de salto
- Tiempo inicial
- Tiempo final
- Gravedad
- Condiciones iniciales

C. Salidas

- Gráfica de posiciones en el tiempo
- Archivo CSV

IV. Diseño

- Modelo: Predicción de trayectoria
- Vista: Menú con instrucciones del programa para asignar valores
- Controlador: Comparación de posiciones



Comportamiento detallado de la ecuación de Euler:

```

float x0 = 10, x1 = 10, x2;
float op1, op2, mass, k, discreet_time, index, time;

points = vars->points;
k = vars->jumps_constant;
mass = vars->mass;
discreet_time = (vars->ending_time - vars->initial_time) / (vars->points);

for(index = 1; index <= points; index++){
    time = index/points;
    op1 = -x0 * (1 + (k * pow(discreet_time, 2)) / mass);
    op2 = 2 * x1 + pow(discreet_time, 2) * GRAVITY;
    x2 = op1 + op2;

    if(x0 <= x2){
        deaths++;
    } else {
        lives++;
    }

    fprintf(file, "%.2f, %.2f\n", time, x2);

    x0 = x1;
    x1 = x2;
}

if(deaths > 1) {
    printf("The person is death\n");
} else{
    printf("The person is still alive\n");
}
  
```

Ejemplo de resultado obtenido por medio de la ecuación de Euler:

```

BUNGEE_FILE_1.CSV
You, 19 minutes ago | 1 author (You)
1 0.02, 8.37
2 0.04, 5.11
3 0.06, 0.74
4 0.08, -3.70
5 0.10, -6.80
6 0.12, -7.15
7 0.14, -3.75
8 0.16, 3.50
9 0.18, 13.52
10 0.20, 23.99
11 0.22, 31.71
12 0.24, 33.31
13 0.26, 26.34
14 0.28, 10.28
15 0.30, -12.64
16 0.32, -37.28
17 0.34, -56.31
18 0.36, -61.84
19 0.38, -47.77
20 0.40, -12.36
21 0.42, 39.02
22 0.44, 97.72
23 0.46, 144.31
24 0.48, 161.21
25 0.50, 133.49
26 0.52, 55.76
27 0.54, -63.13
28 0.56, -198.28
29 0.58, -311.66
30 0.60, -360.03
31 0.62, -307.09
  
```

Script de GNU Plot:

```
void generate_plot(char *name) {
    FILE * new_GNUPlot_Window;
    char file_name[MAX];
    sprintf(file_name, "plot \"%s\" w l", name);

    new_GNUPlot_Window = popen("gnuplot -persist", "w");
    fprintf(new_GNUPlot_Window, "%s\n", file_name);

    //pclose(new_GNUPlot_Window);    You, a few seconds
```

Manual de usuario

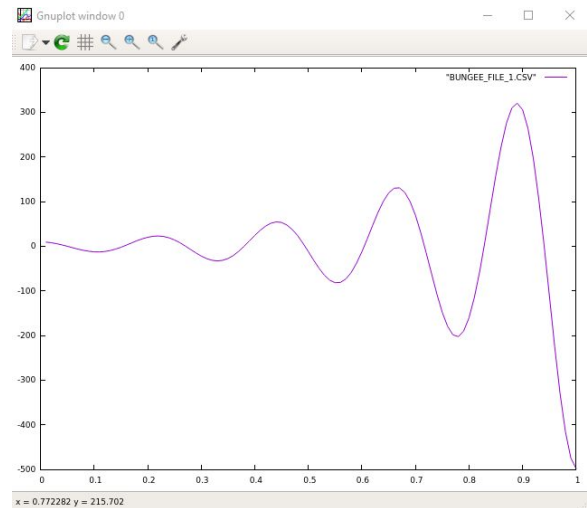
1- Ejecutar el programa

```
~/TDA/practices/EulerMethod/pi$ make
ls: cannot access 'BUNGEE_FILE*.CSV': No such file or directory
gcc -c -o test_bungee.o test_bungee.c -lm -I -Wall
gcc -c -o model.o model.c -lm -I -Wall
gcc -c -o view.o view.c -lm -I -Wall
gcc -c -o controller.o controller.c -lm -I -Wall
gcc -o test_bungee test_bungee.o model.o view.o controller.o -lm -I -Wall
~/TDA/practices/EulerMethod/pi$ ./test_bungee
```

2- Ingresar los valores para predecir la trayectoria de una persona al ser jalada desde el Bungee.

```
WELCOME TO THE BUNGEE JUMPING GAME
Input the mass' value: 60
Input the bungee's jump constant: 500
Input the initial time: 0
Input the ending time: 10
Input the number of points you want to plot: 100
```

3- Gráfica obtenida utilizando el método de Euler.



EulerMethod > p1 > BUNGEE_FILE_1.CSV

1	0.01	9.26
2	0.02	7.79
3	0.03	5.65
4	0.04	2.95
5	0.05	-0.11
6	0.06	-3.33
7	0.07	-6.44
8	0.08	-9.17
9	0.09	-11.27
10	0.10	-12.51
11	0.11	-12.71
12	0.12	-11.76
13	0.13	-9.67
14	0.14	-6.49
15	0.15	-2.41
16	0.16	2.31
17	0.17	7.33
18	0.18	12.25
19	0.19	16.67
20	0.20	20.15
21	0.21	22.35
22	0.22	22.97
23	0.23	21.82
24	0.24	18.85
25	0.25	14.17
26	0.26	8.01
27	0.27	0.77
28	0.28	-7.04
29	0.29	-14.81
30	0.30	-21.91
31	0.31	-27.66
32	0.32	-31.50

EulerMethod > p1 > BUNGEE_FILE_1.CSV

33	0.33	-32.93
34	0.34	-31.64
35	0.35	-27.51
36	0.36	-20.64
37	0.37	-11.38
38	0.38	-0.30
39	0.39	11.82
40	0.40	24.07
41	0.41	35.43
42	0.42	44.88
43	0.43	51.48
44	0.44	54.43
45	0.45	53.20
46	0.46	47.52
47	0.47	37.51
48	0.48	23.64
49	0.49	6.75
50	0.50	-12.03
51	0.51	-31.26
52	0.52	-49.39
53	0.53	-64.83
54	0.54	-76.04
55	0.55	-81.76
56	0.56	-81.04
57	0.57	-73.41
58	0.58	-58.93
59	0.59	-38.23
60	0.60	-12.53
61	0.61	16.46
62	0.62	46.60
63	0.63	75.45
64	0.64	100.53

EulerMethod > p1 > BUNGEE_FILE_1.CSV

65	0.65	119.41
66	0.66	130.02
67	0.67	130.77
68	0.68	120.78
69	0.69	100.00
70	0.70	69.25
71	0.71	30.26
72	0.72	-14.40
73	0.73	-61.48
74	0.74	-107.26
75	0.75	-147.83
76	0.76	-179.35
77	0.77	-198.46
78	0.78	-202.53
79	0.79	-189.06
80	0.80	-160.41
81	0.81	-114.04
82	0.82	-56.00
83	0.83	12.69
84	0.84	86.00
85	0.85	158.43
86	0.86	223.79
87	0.87	276.04
88	0.88	309.75
89	0.89	320.54
90	0.90	305.63
91	0.91	264.10
92	0.92	197.20
93	0.93	108.39
94	0.94	3.24
95	0.95	-110.84
96	0.96	-225.09

97	0.97	-330.01
98	0.98	-416.07
99	0.99	-474.53
100	1.00	-498.23

4- Elegir si se quiere volver a jugar

```
WELCOME TO THE BUNGEE JUMPING GAME
Input the mass' value: 60
Input the bungee's jump constant: 500
Input the initial time: 0
Input the ending time: 10
Input the number of points you want to plot: 100
The person got hit

Do you want to play again?
[y] [n]
Select an option-->
```

V. Conclusión

La práctica a realizar en la materia de Taller de Aplicaciones tiene como objetivo reforzar los temas vistos en clase. De igual modo, es un apoyo que permite al alumno entender los conceptos aplicados en una situación real.

Cabe señalar que las herramientas aprendidas durante clase serán utilizadas en la mayoría de las funciones a desarrollar y que permiten reforzar el conocimiento mediante los diferentes usos a dar en la práctica.

VI. Referencias

[1] Steve S. Skiena. "The Algorithm Design Manual". Second Edition, Springer.