

Simulación Interactiva de Viscosidad - Python

Nombre del estudiante:

Julian David Carrero Ariza, 20201005199

Institución: Universidad Distrital Francisco José de Caldas

Fecha: January 28, 2025

Introducción

El presente documento describe una simulación interactiva de viscosidad desarrollada en Python utilizando la librería `tkinter` para la interfaz gráfica. El modelo físico se basa en las siguientes ecuaciones:

- Fuerza de viscosidad: $F_v = -k_v \cdot V$,
- Aceleración: $a = \frac{F_v}{m}$,
- Velocidad final: $v_f = v + a \cdot t$,
- Desplazamiento: $\Delta pos = v \cdot t + \frac{1}{2} \cdot a \cdot t^2$.

La simulación permite al usuario definir las condiciones iniciales y observar paso a paso cómo se actualizan las posiciones y velocidades.

Código Python

El siguiente es el código completo de la simulación:

```
1 import tkinter as tk
2
3 def calcular_aceleracion(f, m):
4     """Calcula la aceleracion dada la fuerza y la masa."""
5     return [f[0] / m, f[1] / m]
6
7 def un_paso(v, f, t, m):
8     """
9     Realiza un paso de simulacion para calcular el cambio en
10    posicion y velocidad.
11    """
12    a = calcular_aceleracion(f, m) # Calcular aceleracion
13    v_f = [v[0] + a[0] * t, v[1] + a[1] * t] # Calcular
14    velocidad final
```

```

13     delta_pos = [v[0] * t + 0.5 * a[0] * t**2, v[1] * t + 0.5 * a
14                 [1] * t**2]
15     v_f = [0 if abs(vel) < 1e-5 else vel for vel in v_f] #
16         Ajustar pequenas velocidades
17     return delta_pos, v_f
18
19 def acumula_pasos(pos_i, v_i, k_v, pasos, t, m, output_widget):
20     """
21     Simula varios pasos acumulando desplazamiento y actualizando
22     posiciones.
23     """
24     pos = pos_i
25     v = v_i
26     for i in range(pasos):
27         f_visc = [-k_v * v[0], -k_v * v[1]] # Calcular fuerza de
28             viscosidad
29         delta_pos, v = un_paso(v, f_visc, t, m) # Realizar un
30             paso
31         pos = [pos[0] + delta_pos[0], pos[1] + delta_pos[1]] #
32             Actualizar posicion
33         output_widget.insert(tk.END, f"Paso_{i+1}: Posicion={
34             pos}, Velocidad={v}\n")
35         output_widget.yview(tk.END) # Scroll automatico
36
37 # Crear ventana principal
38 window = tk.Tk()
39 window.title("Simulacion de Viscosidad")
40
41 # Entradas para parametros
42 tk.Label(window, text="Posicion_inicial(x,y):").grid(row=0,
43     column=0, sticky="w")
44 pos_input = tk.Entry(window)
45 pos_input.insert(0, "0,0")
46 pos_input.grid(row=0, column=1)
47
48 tk.Label(window, text="Velocidad_inicial(vx,vy):").grid(row=1,
49     column=0, sticky="w")
50 vel_input = tk.Entry(window)
51 vel_input.insert(0, "10,5")
52 vel_input.grid(row=1, column=1)
53
54 tk.Label(window, text="Constante de viscosidad(k_v):").grid(row
55     =2, column=0, sticky="w")
56 kv_input = tk.Entry(window)
57 kv_input.insert(0, "0.5")
58 kv_input.grid(row=2, column=1)
59
60 tk.Label(window, text="Masa(m):").grid(row=3, column=0, sticky="
61     w")
62 m_input = tk.Entry(window)
63 m_input.insert(0, "1")

```

```

53 m_input.grid(row=3, column=1)
54
55 tk.Label(window, text="Tiempo_por_paso(t):").grid(row=4, column
    =0, sticky="w")
56 t_input = tk.Entry(window)
57 t_input.insert(0, "1")
58 t_input.grid(row=4, column=1)
59
60 tk.Label(window, text="Numero_de_pasos:").grid(row=5, column=0,
    sticky="w")
61 steps_input = tk.Entry(window)
62 steps_input.insert(0, "10")
63 steps_input.grid(row=5, column=1)
64
65 # Area de salida
66 tk.Label(window, text="Resultados_de_la_simulacion:").grid(row=6,
    column=0, sticky="w")
67 output = tk.Text(window, height=10, width=50)
68 output.grid(row=7, column=0, columnspan=2, pady=10)
69
70 # Boton para iniciar simulacion
71 def iniciar_simulacion():
72     output.delete(1.0, tk.END) # Limpiar resultados previos
73     pos_i = list(map(float, pos_input.get().split(",")))
74     v_i = list(map(float, vel_input.get().split(",")))
75     k_v = float(kv_input.get())
76     m = float(m_input.get())
77     t = float(t_input.get())
78     pasos = int(steps_input.get())
79     acumula_pasos(pos_i, v_i, k_v, pasos, t, m, output)
80
81 start_button = tk.Button(window, text="Iniciar_Simulacion",
    command=iniciar_simulacion)
82 start_button.grid(row=8, column=0, columnspan=2)
83
84 # Ejecutar ventana
85 window.mainloop()

```

Listing 1: Simulación Interactiva de Viscosidad en Python

Descripción del Código

Este programa utiliza una interfaz gráfica para solicitar los parámetros iniciales de la simulación y presenta los resultados en un cuadro de texto. El modelo matemático incluye la fuerza de viscosidad proporcional a la velocidad, y las posiciones y velocidades se calculan iterativamente.