

Guía de Ejercicios de Python

Profesor: Guillermo Cid Ampuero

Introducción

Esta guía contiene 20 ejercicios de programación en Python orientados al desarrollo de habilidades en programación numérica. Los ejercicios están organizados de manera progresiva, comenzando con conceptos básicos y avanzando hacia métodos numéricos como Euler y Heun. Se requiere el uso de las librerías `numpy` y `matplotlib`.

Librerías Requeridas

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

1. Ejercicios Básicos - Arreglos y Ciclos

1.1. Ejercicio 1: Tabla de Multiplicar

Crear un programa que genere y muestre la tabla de multiplicar del número 7 usando un ciclo `for` y `np.arange`. El programa debe mostrar desde 7×1 hasta 7×12 .

1.2. Ejercicio 2: Suma de Elementos

Crear un arreglo NumPy con los números del 1 al 100 usando `np.arange`. Calcular la suma de todos los elementos usando un ciclo `for` y comparar el resultado con la función `np.sum()`.

1.3. Ejercicio 3: Generación de Secuencias

Generar tres arreglos usando `np.arange`:

- Números pares del 2 al 50
- Números impares del 1 al 49
- Números del 10 al 1 (orden decreciente)

Mostrar cada arreglo y su longitud.

1.4. Ejercicio 4: Operaciones con Arreglos

Crear dos arreglos: uno con valores de 0 a 10 y otro con valores de 10 a 20. Realizar las operaciones de suma, resta, multiplicación y división elemento por elemento. Mostrar los resultados.

2. Ejercicios Intermedios - Funciones y Gráficas

2.1. Ejercicio 5: Función Cuadrática

Programar la función $f(x) = ax^2 + bx + c$ donde $a = 2$, $b = -3$, $c = 1$. Evaluar la función para valores de x desde -5 hasta 5 con paso 0.1. Graficar el resultado usando `matplotlib`.

2.2. Ejercicio 6: Funciones Trigonométricas

Crear un arreglo de valores de x desde 0 hasta 2π con 100 puntos. Calcular $\sin(x)$, $\cos(x)$ y $\tan(x)$. Graficar las tres funciones en la misma figura con diferentes colores y leyendas.

2.3. Ejercicio 7: Comparación de Funciones Exponenciales

Comparar las funciones $y_1 = e^x$, $y_2 = e^{2x}$ y $y_3 = e^{-x}$ para x entre -2 y 2. Graficar las tres funciones y determinar cuál crece más rápido.

2.4. Ejercicio 8: Función Logística

Programar la función logística $P(t) = \frac{K}{1+ Ae^{-rt}}$ donde $K = 100$, $A = 10$, $r = 0,5$. Evaluar para t desde 0 hasta 20. Graficar y analizar el comportamiento de crecimiento.

3. Ejercicios Avanzados - Aplicaciones Físicas

3.1. Ejercicio 9: Caída Libre

Programar la ecuación de caída libre: $h(t) = h_0 - \frac{1}{2}gt^2$ donde $h_0 = 100$ m y $g = 9,81$ m/s². Calcular la altura para tiempos desde 0 hasta el tiempo que tarda en tocar el suelo. Graficar altura vs tiempo.

3.2. Ejercicio 10: Velocidad en Caída Libre

Extender el ejercicio anterior calculando también la velocidad: $v(t) = -gt$. Graficar en la misma figura la posición y velocidad vs tiempo usando diferentes ejes y (subplot).

3.3. Ejercicio 11: Descarga de Capacitor

Programar la ecuación de descarga de un capacitor: $Q(t) = Q_0 e^{-\frac{t}{RC}}$ donde $Q_0 = 1000$ C, $R = 10000$, $C = 100 \times 10^{-6}$ F. Graficar la carga vs tiempo durante 10 constantes de tiempo.

3.4. Ejercicio 12: Corriente en Circuito RC

Calcular la corriente durante la descarga del capacitor: $I(t) = -\frac{Q_0}{RC} e^{-\frac{t}{RC}}$. Graficar carga y corriente en función del tiempo en subplots separados.

3.5. Ejercicio 13: Movimiento Armónico Simple

Programar las ecuaciones: $x(t) = A \cos(\omega t + \phi)$ y $v(t) = -A\omega \sin(\omega t + \phi)$ donde $A = 5$ m, $\omega = 2$ rad/s, $\phi = \pi/4$. Graficar posición y velocidad vs tiempo.

3.6. Ejercicio 14: Péndulo Simple (Aproximación)

Para ángulos pequeños, $\theta(t) = \theta_0 \cos(\sqrt{\frac{g}{L}}t)$ donde $\theta_0 = 0,2$ rad, $L = 1$ m. Calcular el período y graficar el movimiento durante 3 períodos completos.

3.7. Ejercicio 15: Ley de Enfriamiento de Newton

Programar $T(t) = T_{amb} + (T_0 - T_{amb})e^{-kt}$ donde $T_{amb} = 20^\circ\text{C}$, $T_0 = 100^\circ\text{C}$, $k = 0,1$ min⁻¹. Graficar la temperatura vs tiempo hasta que alcance 25°C.

4. Ejercicios de Métodos Numéricos

4.1. Ejercicio 16: Método de Euler - Ecuación Simple

Implementar el método de Euler para resolver $\frac{dy}{dt} = -2y$ con $y(0) = 1$. Usar $h = 0,1$ y calcular desde $t = 0$ hasta $t = 2$. Comparar con la solución analítica $y(t) = e^{-2t}$.

Algoritmo del Método de Euler:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n) \quad (1)$$

donde $f(t, y) = \frac{dy}{dt}$.

4.2. Ejercicio 17: Método de Euler - Crecimiento Poblacional

Usar Euler para resolver $\frac{dP}{dt} = rP(1 - \frac{P}{K})$ con $P(0) = 10$, $r = 0,1$, $K = 1000$. Calcular desde $t = 0$ hasta $t = 50$ con $h = 0,5$. Graficar el resultado.

4.3. Ejercicio 18: Método de Heun - Introducción

Implementar el método de Heun para resolver $\frac{dy}{dt} = y - t$ con $y(0) = 1$. Usar $h = 0,1$ desde $t = 0$ hasta $t = 1$.

Método de Heun (Explicación): El método de Heun es una mejora del método de Euler que proporciona mayor precisión. Se basa en usar dos evaluaciones de la derivada:

Paso 1: Predictor (como Euler)

$$\tilde{y}_{n+1} = y_n + h \cdot f(t_n, y_n) \quad (2)$$

Paso 2: Corrector

$$y_{n+1} = y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})] \quad (3)$$

El método promedia las pendientes al inicio y al final del intervalo, mejorando la aproximación.

4.4. Ejercicio 19: Método de Heun - Sistema Masa-Resorte

Usar Heun para resolver el sistema de ecuaciones del oscilador armónico:

$$\frac{dx}{dt} = v \quad (4)$$

$$\frac{dv}{dt} = -\omega^2 x \quad (5)$$

con $x(0) = 1$, $v(0) = 0$, $\omega = 2$. Calcular desde $t = 0$ hasta $t = 2\pi$ con $h = 0,01$.

4.5. Ejercicio 20: Comparación Euler vs Heun

Resolver la ecuación $\frac{dy}{dt} = -\cos(t) + \sin(y)$ con $y(0) = 1$ usando ambos métodos (Euler y Heun) con $h = 0,1$ desde $t = 0$ hasta $t = 5$. Graficar ambas soluciones en la misma figura y comparar los resultados. Discutir cuál método es más preciso.