UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSÍ

Facultad de Estudios Profesionales de la Zona Media

Materia: Matemáticas por Computadora

Profesor:Ing. Luis Padrón

Alumno: Jimena Balderas Coronado

Práctica 8

Fecha: 8 de abril de 2025

1 Introduccion

La integración de Romberg es una técnica numérica basada en la regla compuesta del trapecio que utiliza el proceso de extrapolación de Richardson para mejorar la precisión en la aproximación de una integral definida.

Se define una secuencia de aproximaciones $R_{k,j}$ hacia el valor exacto de la integral.

Desarrollo

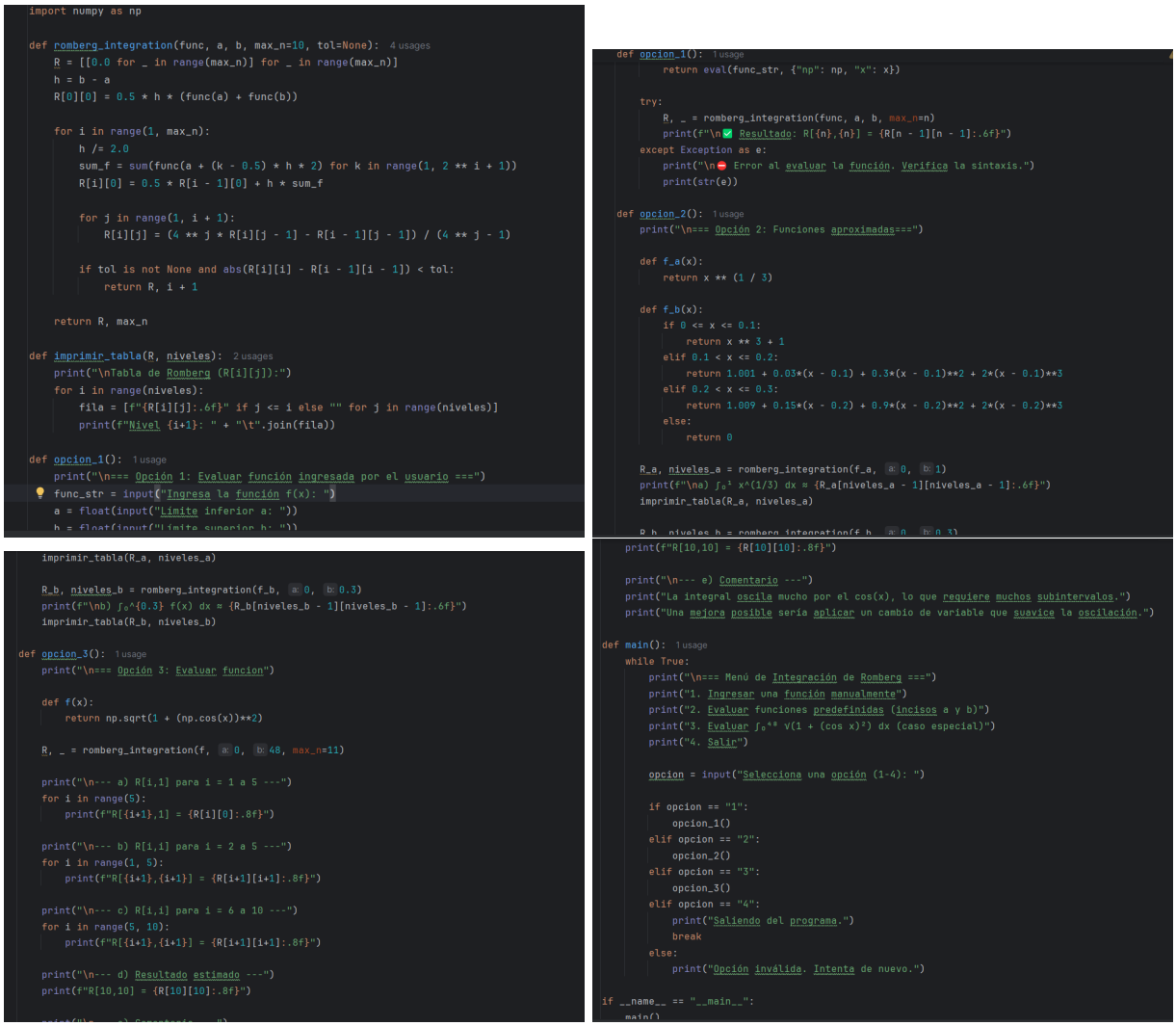
Explicación del código para resolver los problemas

Para empezar el código, se debe importar la librería `numpy` que es la que nos permite ingresar funciones, y lo guardaremos como `np`.

Para realizar la integración de Romberg, primero crearemos una tabla donde se tendrá el valor máximo de R y se guardarán los valores de aproximación. Para crear la aproximación, se usará la regla del trapecio simple.

Posteriormente, comienza un bucle de refinamiento y se reduce el tamaño de h para cada iteración. Se suman los nuevos puntos intermedios del intervalo que no se evaluaron y se actualiza la nueva fila de la tabla. Finalmente, se imprime la tabla.

Al ser varias opciones que se pueden hacer con el código, se contiene un menú. Para la primera opción, el usuario escribe una función, y se calcula la integral. Para la segunda opción, se calcula la integral dada y se evalúa la función por tramos. Finalmente, como tercera opción se calcula la integral dada y se muestran distintos subconjuntos de la tabla de Romberg: una con aproximaciones por regla del trapecio y la otra con aproximaciones con corrección de errores.



```python
import numpy as np

def romberg_integration(func, a, b, max_n=10, tol=None):  4 usages
    R = [[0.0 for _ in range(max_n)] for _ in range(max_n)]
    h = b - a
    R[0][0] = 0.5 * h * (func(a) + func(b))

    for i in range(1, max_n):
        h /= 2.0
        sum_f = sum(func(a + (k - 0.5) * h * 2) for k in range(1, 2 ** i + 1))
        R[i][0] = 0.5 * R[i - 1][0] + h * sum_f

        for j in range(1, i + 1):
            R[i][j] = (4 ** j * R[i][j - 1] - R[i - 1][j - 1]) / (4 ** j - 1)

        if tol is not None and abs(R[i][i] - R[i - 1][i - 1]) < tol:
            return R, i + 1

    return R, max_n

def imprimir_tabla(R, niveles):  2 usages
    print("\nTabla de Romberg (R[i][j]):")
    for i in range(niveles):
        fila = [f"{R[i][j]:.6f}" if j <= i else "" for j in range(niveles)]
        print(f"Nivel {i+1}: " + "\t".join(fila))

def opcion_1():  1 usage
    print("\n=== Opción 1: Evaluar función ingresada por el usuario ===")
    func_str = input("Ingresa la función f(x): ")
    a = float(input("Límite inferior a: "))
    b = float(input("Límite superior b: "))
```

```python
def opcion_1():  1 usage
        return eval(func_str, {"np": np, "x": x})

    try:
        R, _ = romberg_integration(func, a, b, max_n=n)
        print(f"\n✅ Resultado: R[{n},{n}] = {R[n - 1][n - 1]:.6f}")
    except Exception as e:
        print("\n⛔ Error al evaluar la función. Verifica la sintaxis.")
        print(str(e))

def opcion_2():  1 usage
    print("\n=== Opción 2: Funciones aproximadas===")

    def f_a(x):
        return x ** (1 / 3)

    def f_b(x):
        if 0 <= x <= 0.1:
            return x ** 3 + 1
        elif 0.1 < x <= 0.2:
            return 1.001 + 0.03*(x - 0.1) + 0.3*(x - 0.1)**2 + 2*(x - 0.1)**3
        elif 0.2 < x <= 0.3:
            return 1.009 + 0.15*(x - 0.2) + 0.9*(x - 0.2)**2 + 2*(x - 0.2)**3
        else:
            return 0

    R_a, niveles_a = romberg_integration(f_a,  a: 0,  b: 1)
    print(f"\na) ∫₀¹ x^(1/3) dx ≈ {R_a[niveles_a - 1][niveles_a - 1]:.6f}")
    imprimir_tabla(R_a, niveles_a)

    R_b, niveles_b = romberg_integration(f_b,  a: 0,  b: 0.3)
```

```python
    imprimir_tabla(R_a, niveles_a)

    R_b, niveles_b = romberg_integration(f_b,  a: 0,  b: 0.3)
    print(f"\nb) ∫₀^0.3 f(x) dx ≈ {R_b[niveles_b - 1][niveles_b - 1]:.6f}")
    imprimir_tabla(R_b, niveles_b)

def opcion_3():  1 usage
    print("\n=== Opción 3: Evaluar funcion")

    def f(x):
        return np.sqrt(1 + (np.cos(x))**2)

    R, _ = romberg_integration(f,  a: 0,  b: 48, max_n=11)

    print("\n--- a) R[i,1] para i = 1 a 5 ---")
    for i in range(5):
        print(f"R[{i+1},1] = {R[i][0]:.8f}")

    print("\n--- b) R[i,i] para i = 2 a 5 ---")
    for i in range(1, 5):
        print(f"R[{i+1},{i+1}] = {R[i+1][i+1]:.8f}")

    print("\n--- c) R[i,i] para i = 6 a 10 ---")
    for i in range(5, 10):
        print(f"R[{i+1},{i+1}] = {R[i+1][i+1]:.8f}")

    print("\n--- d) Resultado estimado ---")
    print(f"R[10,10] = {R[10][10]:.8f}")
```

```python
    print(f"R[10,10] = {R[10][10]:.8f}")

    print("\n--- e) Comentario ---")
    print("La integral oscila mucho por el cos(x), lo que requiere muchos subintervalos.")
    print("Una mejora posible sería aplicar un cambio de variable que suavice la oscilación.")

def main():  1 usage
    while True:
        print("\n=== Menú de Integración de Romberg ===")
        print("1. Ingresar una función manualmente")
        print("2. Evaluar funciones predefinidas (incisos a y b)")
        print("3. Evaluar ∫₀^48 √(1 + (cos x)²) dx (caso especial)")
        print("4. Salir")

        opcion = input("Selecciona una opción (1-4): ")

        if opcion == "1":
            opcion_1()
        elif opcion == "2":
            opcion_2()
        elif opcion == "3":
            opcion_3()
        elif opcion == "4":
            print("Saliendo del programa.")
            break
        else:
            print("Opción inválida. Intenta de nuevo.")

if __name__ == "__main__":
    main()
```

Figure 1: Código de ejemplo para integración de Romberg

Ejercicios

1.1 Ejercicio 1

Por medio de la integración de Romberg calcule $R_{3,3}$ para las siguientes funciones.

a. $\int_1^{1.5} x^2 \ln x \, dx$

b. $\int_0^1 x^2 e^{-x} \, dx$

c. $\int_0^{0.35} \frac{2}{x^2 - 4} \, dx$

d. $\int_0^{\pi/4} x^2 \operatorname{sen} x \, dx$

e. $\int_0^{\pi/4} e^{3x} \operatorname{sen} 2x \, dx$

f. $\int_1^{1.6} \frac{2x}{x^2 - 4} \, dx$

g. $\int_3^{3.5} \frac{x}{\sqrt{x^2 - 4}} \, dx$

h. $\int_0^{\pi/4} (\cos x)^2 \, dx$

Figure 2: Funciones para calcular $R_{3,3}$

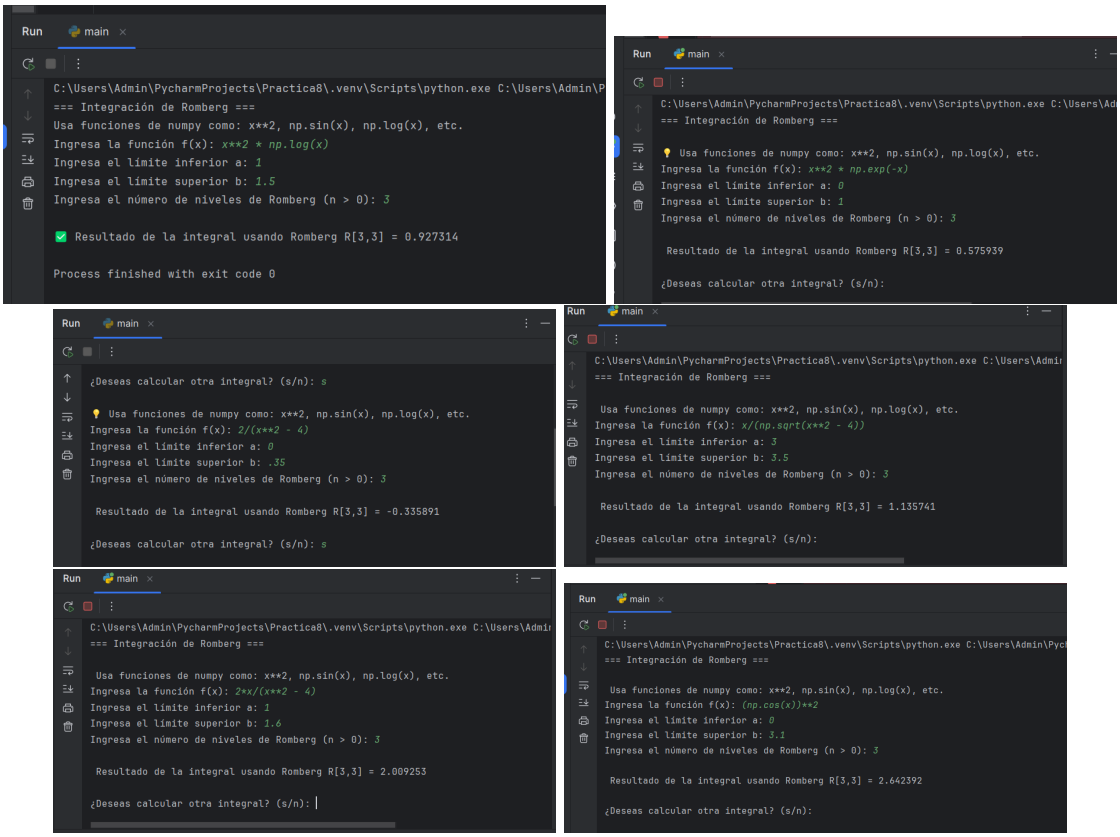


Figure 3: Resultados del Ejercicio 1

1.2 Ejercicio 2

Calcule $R_{4,4}$ para las mismas integrales.

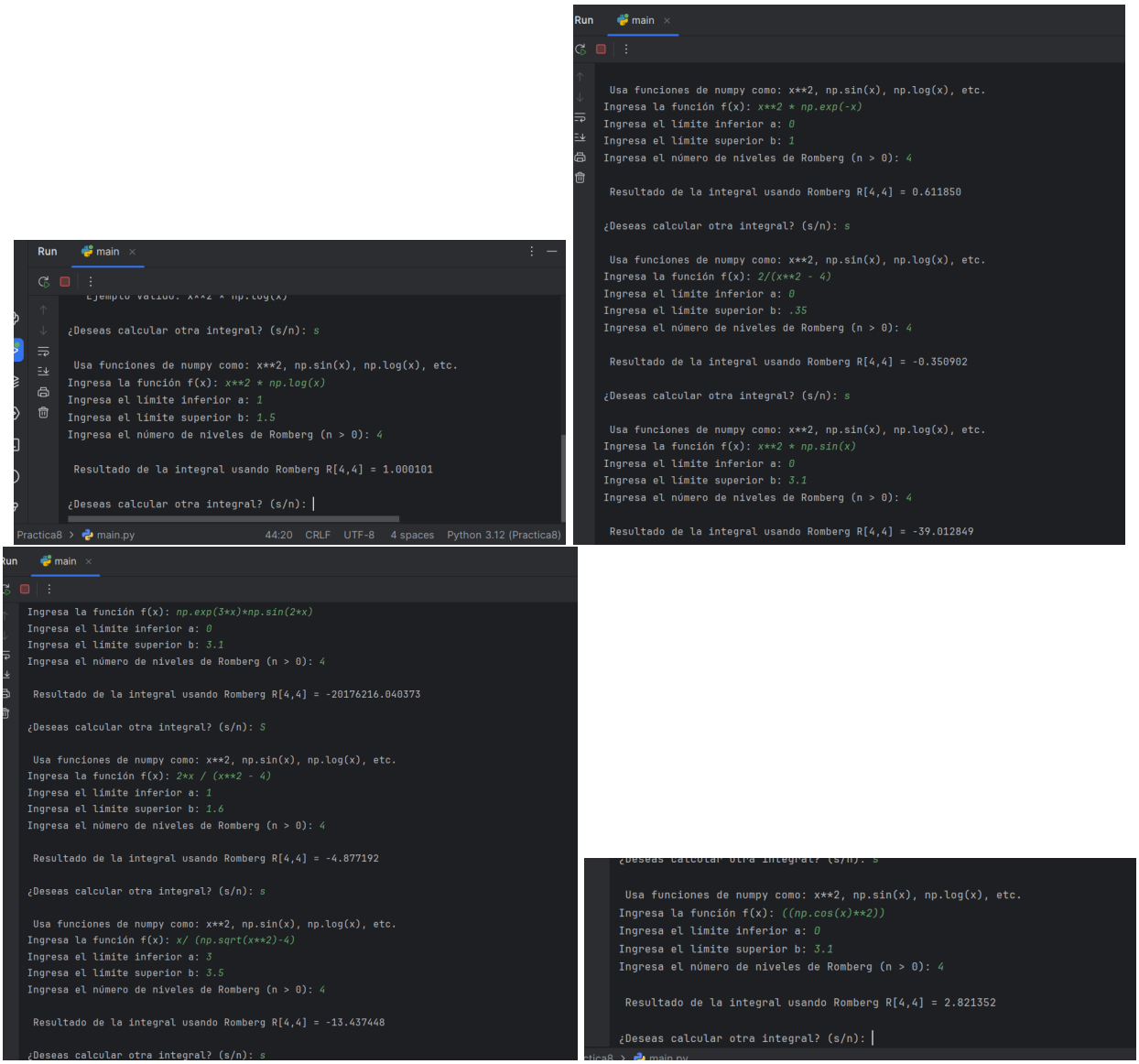


Figure 4: Resultados del Ejercicio 2

Ejercicio 4

Aplique la integración de Romberg a las siguientes integrales hasta que $R_{n-1,n-1}$ y $R_{n,n}$ concuerden con una exactitud de 10^{-4}.

4. Aplique la integración de Romberg a las siguientes integrales hasta que $R_{n-1,\,n-1}$ y $R_{n,n}$ concuerden con una exactitud de 10^{-4}.

a. $\int_0^1 x^{1/3}\,dx$

b. $\int_0^{0.3} f(x)\,dx$, donde

$$f(x) = \begin{cases} x^3 + 1, & 0 \le x \le 0.1 \\ 1.001 + 0.03(x - 0.1) + 0.3(x - 0.1)^2 + 2(x - 0.1)^3, & 0.1 < x \le 0.2, \\ 1.009 + 0.15(x - 0.2) + 0.9(x - 0.2)^2 + 2(x - 0.2)^3, & 0.2 < x \le 0.3. \end{cases}$$

Figure 5: Integrales para exactitud de 10^{-4}

```
a) ∫₀¹ x^(1/3) dx ≈ 1.888508

Tabla de Romberg (R[i][j]):
Nivel 1: 0.500000
Nivel 2: 1.219207   1.458943
Nivel 3: 1.564806   1.680006   1.694743
Nivel 4: 1.731576   1.787166   1.794310   1.795890
Nivel 5: 1.812448   1.839406   1.842889   1.843660   1.843847
Nivel 6: 1.851857   1.864994   1.866700   1.867078   1.867169   1.867192
Nivel 7: 1.871147   1.877577   1.878415   1.878601   1.878647   1.878658   1.878661
Nivel 8: 1.880625   1.883784   1.884198   1.884290   1.884312   1.884318   1.884319   1.884319
Nivel 9: 1.885297   1.886855   1.887059   1.887105   1.887116   1.887119   1.887119   1.887120   1.887120
Nivel 10: 1.887607  1.888377   1.888478   1.888501   1.888506   1.888508   1.888508   1.888508   1.888508   1.888508

b) ∫₀^0.3 f(x) dx ≈ 0.302425

Tabla de Romberg (R[i][j]):
Nivel 1: 0.305250
Nivel 2: 0.303150   0.302450
Nivel 3: 0.302607   0.302427   0.302425
Nivel 4: 0.302471   0.302425   0.302425   0.302425
Nivel 5: 0.302436   0.302425   0.302425   0.302425   0.302425
Nivel 6: 0.302428   0.302425   0.302425   0.302425   0.302425   0.302425
Nivel 7: 0.302426   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425
Nivel 8: 0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425
Nivel 9: 0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425
Nivel 10: 0.302425  0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425   0.302425
```

Figure 6: Resultados del Ejercicio 4

Ejercicio 10

Use la integración de Romberg para calcular las siguientes aproximaciones a π:

10. Use la integración de Romberg para calcular las siguientes aproximaciones a

$$\int_0^{48} \sqrt{1 + (\cos x)^2}\,dx.$$

[*Nota*: Los resultados de este ejercicio son muy interesantes en caso de que esté usted utilizando un dispositivo que maneje una aritmética entre siete y nueve dígitos.]

a. Determine $R_{1,1}$, $R_{2,1}$, $R_{3,1}$, $R_{4,1}$ y $R_{5,1}$ y utilice estas aproximaciones para predecir el valor de la integral.

b. Determine $R_{2,2}$, $R_{3,3}$, $R_{4,4}$ y $R_{5,5}$, y modifique su predicción.

c. Determine $R_{6,1}$, $R_{6,2}$, $R_{6,3}$, $R_{6,4}$, $R_{6,5}$ y $R_{6,6}$ y modifique su predicción.

d. Determine $R_{7,7}$, $R_{8,8}$, $R_{9,9}$ y $R_{10,10}$ y haga una predicción final.

e. Explique por qué esta integral causa problemas en la integración de Romberg y cómo podemos reformularla para obtener más fácilmente una aproximación exacta.

Figure 7: Aproximaciones a π con Romberg

```
--- a) R[i,1] para i = 1 a 5 ---
R[1,1] = 62.43737140
R[2,1] = 90.67853693
R[3,1] = 104.22215831
R[4,1] = 110.91068560
R[5,1] = 114.23468060

--- b) R[i,i] para i = 2 a 5 ---
R[2,2] = 109.31299477
R[3,3] = 113.49917008
R[4,4] = 115.53007540
R[5,5] = 115.40527808

--- c) R[i,i] para i = 6 a 10 ---
R[6,6] = 116.20324227
R[7,7] = 116.44711298
R[8,8] = 116.57260039
R[9,9] = 116.63552220
R[10,10] = 116.66698144

--- d) Resultado estimado ---
R[10,10] = 116.66698144

--- e) Comentario ---
La integral oscila mucho por el cos(x), lo que requiere muchos subintervalos.
Una mejora posible sería aplicar un cambio de variable que suavice la oscilación.
```

Figure 8: Resultados del Ejercicio 10

Conclusiones

Tuve algunos problemas en la comprensión del tema, pero me dirigí a materiales de apoyo como YouTube para comprender la integración de Romberg. De igual manera, ciertas herramientas de internet me hicieron posible concretar el código y obtener los resultados.