Universidad de Guanajuato

Tarea Integrales

Métodos numéricos

Danae Govea Mendoza

15 de marzo de 2024

Segunda y tercera parte

Parte dos.

Resuelve las siguientes integrales definidas:

a) $\int_0^2 3x^2 dx$

b) $\int_0^1 e^x dx$

c) $\int_1^e \frac{1}{x} dx$

d) $\int_{-1}^1 \left(x+2x^2-x^3+5x^4\right) dx$

e) $\int_2^3 \frac{1}{\sqrt{x-1}} dx$

f) $\int_1^2 \frac{2x+1}{x^2+x} dx$

g) $\int_0^{2\pi} senx dx$

h) $\int_0^1 \frac{1}{1+x^2} dx$

i) $\int_2^5 \frac{1}{(x-1)\cdot(x+2)} dx$

j) $\int_0^1 \frac{x}{1+x^4} dx$

1.
```
[9]: #Ejercicio a)

func = lambda x: 3*x**2
a = 0
b = 2

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

8.0
```

2.
```
[14]: #Ejercicio b)

func = lambda x: np.exp(x)
a = 0
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

1.717896378007504
```

3.

```
[16]: #Ejercicio c)

func = lambda x: 1/x
a = np.exp(1)
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

-0.995067271765676
```

4.

```
[3]: #Ejercicio e)

func = lambda x: x + 2*x**2 - x**3 + 5*x**4
a = -1
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

2.444444444444444
```

5.

```
[18]: #Ejercicio e)

func = lambda x: 1 / (np.sqrt(x - 1))
a = 2
b = 3

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

0.8281527373602195
```

6.

```
[20]: #Ejercicio f)

func = lambda x: (2*x + 1)/(x**2 + x)
a = 1
b = 2

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

1.0977130977130978
```

7.

```
[22]: #Ejercicio g)

func = lambda x: np.sin(x)
a = 0
b = 2*np.pi

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

-3.487868498008632e-16
```

8.
```
[23]: #Ejercicio h)

func = lambda x: 1/(1 + x**2)
a = 0
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

0.7868852459016393
```

9.
```
[24]: #Ejercicio i)

func = lambda x: 1/((x - 1)*(x + 2))
a = 2
b = 5

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

0.26810477657935283
```

10.
```
[26]: #Ejercicio j)

func = lambda x: x/(1 + x**4)
a = 0
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

0.38978345363686845
```

Parte tres.

Calcula el área de la región limitada por las siguientes gráficas:

a) $\left.\begin{array}{l} y = x+1 \\ y = 0 \ (EJE\ OX) \\ x = 0 \\ x = 1 \end{array}\right\}$

b) $\left.\begin{array}{l} y = x^2+1 \\ y = 0 \ (EJE\ OX) \\ x = 1 \\ x = 2 \end{array}\right\}$

c) $\left.\begin{array}{l} y = x^3 \\ y = 0 \ (EJE\ OX) \\ x = 0 \\ x = 2 \end{array}\right\}$

d) $\left.\begin{array}{l} y = x^2 \\ y = -x+2 \\ y = 0 \ (EJE\ OX) \end{array}\right\}$

e) $\left.\begin{array}{l} y = x^2-x-2 \\ y = 0 \ (EJE\ OX) \\ x = 0 \\ x = 1 \end{array}\right\}$

f) $\left.\begin{array}{l} y = \cos x \\ y = 0 \ (EJE\ OX) \\ x = \pi/2 \\ x = 3\pi/2 \end{array}\right\}$

g) $\left.\begin{array}{l} y = x^2 \\ y = x \end{array}\right\}$

h) $\left.\begin{array}{l} y = -x^2+6x \\ y = x^2-2x \end{array}\right\}$

1.

```
[29]: # Parte 3
# Calcular el área de la región limitada por las gráficas

#Ejercicio a)
#Utilizando el método Gauss-Legendre

func = lambda x: x + 1
a = 0
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)
```

```
1.5
```

2.

```
[31]: #Ejercicio b)

func = lambda x: x**2 + 1
a = 1
b = 2

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)
```

```
3.333333333333333
```

3.

```
[32]: #Ejercicio c)

func = lambda x: x**3
a = 0
b = 2

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)
```

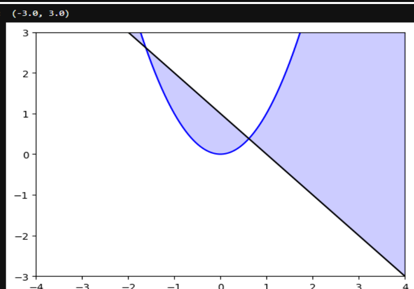
```
3.9999999999999996
```

4.

```
[34]: #Ejercicio c)

func = lambda x: x**2 + x - 2
a = -2
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)
```

```
-4.5
```

```
[33]: (-3.0, 3.0)
```

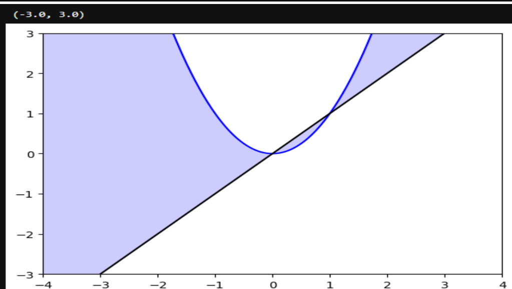


5.
```
[35]: #Ejercicio e)

func = lambda x: x**2 - x - 2
a = 0
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

-2.1666666666666665
```

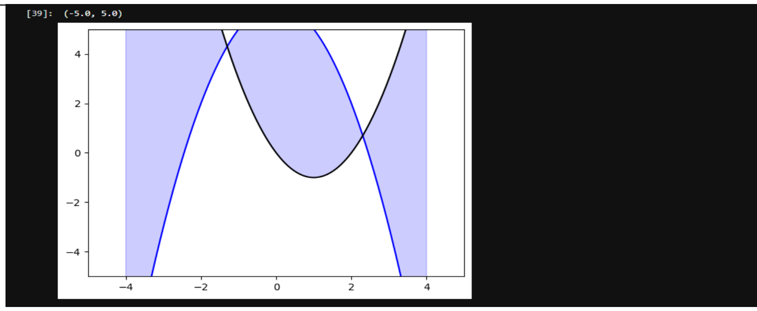
6.
```
[36]: #Ejercicio e)

func = lambda x: np.cos(x)
a = np.pi/2
b = (3*np.pi)/2

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

-1.9358195746511375
```

7.
```
[38]: #Ejercicio e)

func = lambda x: x**2 - x
a = 0
b = 1

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

-0.16666666666666666
```

```
[37]: (-3.0, 3.0)
```



8.
```
[40]: #Ejercicio e)

func = lambda x: -2*x**2 + 2*x + 6
a = 0
b = 2

x = np.array([np.sqrt(1/3), -np.sqrt(1/3)]) #valor de los nodos x_i
w = np.array([1, 1]) #pesos

#cambio de variable
u = (b-a)*x/2 + (a + b)/2

#hacer la evaluación
integral = (b - a) * np.sum(w*func(u))/2
print(integral)

10.666666666666666
```


[39]: (-5.0, 5.0)
4
2
0
−2
−4
−4
−2
0
2
4