Resultados obtenidos para las integrales de la segunda parte de: Tarea de integrales

a)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 2.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  7.999999999999998
```

b)

```
    cj = 2.0 / ((1.0 - xroot**2) * (dlegendre(N, xroot)**2))
    return xroot, cj # Devuelve nodos y pesos

# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  1.7182818275260778
```

c)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [1., np.exp(1)] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  0.9999812926343608
```

d)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [-1., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  3.3333333333333313
```

e)

```python
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [2., 3.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  0.8284269460583165
```

f)

```python
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [1., 2.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  1.098611519048411
```

g)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., np.pi] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  1.9999842284577227
```

h)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  1.3333333333333333
```

i)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [2., 5.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  7.15799062988027
```

j)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  0.3927416219319037
```

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

Resultados para las áreas de la parte 3 de la tarea de integrales

a)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  1.5
```

b)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [1., 2.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  3.3333333333333335
```

c)

```python
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros limites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 2.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  3.9999999999999987
```

d)

Para este caso, encontramos los puntos donde y=x^2 y y=-x+2 se intersectan, estos son 1 y 2, por tanto esos son nuestros limites de integración y la función que definimos será –x+2-x^2

```python
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros limites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [1., 2.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  -1.8333333333333335
```

e)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  -2.1666666666666665
```

f)

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [np.pi/2, 3*np.pi/2] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  -1.9999842284577225
```

g)

Para encontrar nuestros limites de integración, encontramos los puntos de intersección de las curvas, en este caso las curvas y=x^2 y y=x se intersectan cuando x^2=x, si resolvemos esta ecuación nos damos cuenta que las soluciones deben ser 0 y 1 por tanto estos son nuestros puntos de intersección. Ahora la función que pondremos en el programa será x-x^2.

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 1.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  0.16666666666666669
```

h)

Aplicamos la misma idea para este caso, como no tenemos puntos de intersección dados por el problema, tomamos las funciones que nos dan y las igualamos para encontrar los puntos de intersección que serán nuestros limites de integración. Entonces: las gráficas y= -x^2+6x y y= x^2-2x se intersectan en –x^2+6x = x^2-2x, si resolvemos esta ecuación nos damos cuenta que la satisfacen las x: x=0 y x=2, por tanto estos son nuestros puntos de intersección y a su vez los limites de integración que tenemos que poner en el programa, asi como nuestra función que ahora es (-x^2+6x) – (x^2-2x).

```
# Calculamos la integral en los intervalos dados
def gauInt(f, interv, Npts, delt=0.2, Nit=1000, error='dist', eps=1e-05):
    a, b = min(interv), max(interv) # Nuestros límites de integración

    # Utilizamos los nodos y los pesos en estas nuevas variables
    xs, cs = gau_param(Npts, delt=delt, Nit=Nit, error=error, eps=eps)

    # Aplicando la fórmula, cambio de variables
    coeffp = 0.5*(b+a)
    coeffm = 0.5*(b-a)

    ts = coeffp + coeffm*xs # Nueva función t
    fk = cs*f(ts) # Calculamos los valores de la función en los nodos y multiplicamos por los pesos
    val = coeffm*np.sum(fk) # Hacemos la suma de los valores y multiplicamos por el coeficiente
    return val # El resultado de nuestra integral

interv = [0., 2.] # Intervalo
val = gauInt(f, interv, N, eps=1e-11)
print('El resultado es ', val)
```

```
El resultado es  10.666666666666666
```

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