LinAlg-Sympy

December 7, 2020

Tarea Nro. 3 - LinAlg + Sympy

- Nombre y apellido: Ivo Andrés Astudillo
- Fecha: 7 de diciembre de 2020

1 Producto punto

1.1 Use un producto punto y la lista de compras de la tabla 9.4 para determinar su cuenta total en la tienda.

Tabla 9.4 Lista de compras

Artículo	Número necesario	Costo
Leche	2 galones	$3.50 por galón
Huevos	1 docena	$1.25 por docena
Cereal	2 cajas	$4.25 por caja
Sopa	5 latas	$1.55 por lata
Galletas	1 paquete	$3.15 por paquete

[32]:
```
import numpy as np
```

[33]:
```
articulo = np.array([2, 1, 2, 5, 1])
costo = np.array([3.50, 1.25, 4.25, 1.55, 3.15])
print("Cuenta total:", np.dot(articulo, costo))
```

```
Cuenta total: 27.65
```

2 Multiplicación matricial

2.1 Con un calorímetro de bomba se realizó una serie de experimentos. En cada experimento se usó una cantidad diferente de agua. Calcule la capacidad calorífica total para el calorímetro en cada uno de los experimentos, mediante multiplicación matricial, los datos de la tabla 9.8 y la información acerca de la capacidad calorífica que sigue a la tabla.

Tabla 9.8 Propiedades térmicas de un calorímetro de bomba

Experimento núm.	Masa de agua	Masa de acero	Masa de aluminio
1	110 g	250 g	10 g
2	100 g	250 g	10 g
3	101 g	250 g	10 g
4	98.6 g	250 g	10 g
5	99.4 g	250 g	10 g

Componente	Capacidad calorífica
Acero	0.45 J/gK
Agua	4.2 J/gK
Aluminio	0.90 J/gK

[34]:
```
bomba = np.array([
    [110, 250, 10],
    [100, 250, 10],
    [101, 250, 10],
    [98.6, 250, 10],
    [99.4, 250, 10]
]).reshape(5, 3)

capacidad_calorica = np.array([4.2, 0.45, 0.90]).reshape(3, 1)

capacidad_calorica_total = np.dot(bomba, capacidad_calorica)
```

```
i=1
for c in capacidad_calorica_total:
    print(f'Capacidad calórica del experimento {i} = {float(c)}')
    i+=1
```

```
Capacidad calórica del experimento 1 = 583.5
Capacidad calórica del experimento 2 = 541.5
Capacidad calórica del experimento 3 = 545.7
Capacidad calórica del experimento 4 = 535.62
Capacidad calórica del experimento 5 = 538.98
```

3 Determinantes e inversos

3.1 Recuerde que no todas las matrices tienen inverso. Una matriz es singular (es decir: no tiene inverso) si su determinante es igual a 0 (es decir, $|A| = 0$). Use la función determinante para probar si cada una de las siguientes matrices tiene inverso:

$$A = \begin{bmatrix} 2 & -1 \\ 4 & 5 \end{bmatrix}, \qquad B = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 2 \\ 5 & -4 & 0 \end{bmatrix}$$

Si existe un inverso, calcúlelo.

[35]:
```
def detInv(m, nombre):
    det = np.linalg.det(m)
    if det != 0:
        inv = np.linalg.inv(m)
        print(f'El determinante de la matriz {nombre} es: {det}')
        print(f'El inverso de la matriz {nombre} es:')
        print(inv)
        print('\n')
    else:
        print(f'El determinante de la matriz {nombre} es: {det} por lo tanto no␣
 ↪tiene inverso.')
        print('\n')


A = np.array([
    [2, -1],
    [4, 5]
]).reshape(2,2)

B = np.array([
```

```
    [4, 2],
    [2, 1]
]).reshape(2,2)


C = np.array([
    [2, 0, 0],
    [1, 2, 2],
    [5, 4, 0]
]).reshape(3, 3)

detInv(A, 'A')
detInv(B, 'B')
detInv(C, 'C')
```

```
El determinante de la matriz A es: 14.000000000000004
El inverso de la matriz A es:
[[ 0.35714286  0.07142857]
 [-0.28571429  0.14285714]]


El determinante de la matriz B es: 0.0 por lo tanto no tiene inverso.


El determinante de la matriz C es: -15.999999999999998
El inverso de la matriz C es:
[[ 0.5    0.     0.   ]
 [-0.625 -0.     0.25 ]
 [ 0.375  0.5   -0.25 ]]
```

4 Resolución de sistemas ecuaciones lineales

4.1 Resuelva el siguiente sistema de ecuaciones

$$
\begin{aligned}
3x_1 &+4x_2 &+2x_3 &-x_4 &+x_5 &+7x_6 &+x_7 &= 42 \\
2x_1 &-2x_2 &+3x_3 &-4x_4 &+5x_5 &+2x_6 &+8x_7 &= 32 \\
x_1 &+2x_2 &+3x_3 &+x_4 &+2x_5 &+4x_6 &+6x_7 &= 12 \\
5x_1 &+10x_2 &+4x_3 &+3x_4 &+9x_5 &-2x_6 &+x_7 &= -5 \\
3x_1 &+2x_2 &-2x_3 &-4x_4 &-5x_5 &-6x_6 &+7x_7 &= 10 \\
-2x_1 &+9x_2 &+x_3 &+3x_4 &-3x_5 &+5x_6 &+x_7 &= 18 \\
x_1 &-2x_2 &-8x_3 &+4x_4 &+2x_5 &+4x_6 &+5x_7 &= 17
\end{aligned}
$$

[36]:
```
M = np.array([
    [3, 4, 2, -1, 1, 7, 1],
    [2, -2, 3, -4, 5, 2, 8],
    [1, 2, 3, 1, 2, 4, 6],
    [5, 10, 4, 3, 9, -2, 1],
    [3, 2, -2, -4, -5, -6, 7],
    [-2, 9, 1, 3, -3, 5, 1],
    [1, -2, -8, 4, 2, 4, 5],
]).reshape(7, 7)

res = np.array([42, 32, 12, -5, 10, 18, 17]).reshape(7, 1)

print('Solución del sistema de ecuaciones:\n')
print(np.linalg.solve(M, res))
```

```
Solución del sistema de ecuaciones:

[[-0.18899493]
 [ 2.54589061]
 [-3.28057396]
 [-6.75778176]
 [ 1.32124449]
 [ 4.31944831]
 [ 0.62940585]]
```

5 Cálculo

La capacidad calorífica Cp de un gas se puede modelar con la ecuación empírica

$$C_p = a + bT + cT^2 + dT^3 \tag{1}$$

donde a, b, c y d son constantes empíricas y T es la temperatura en grados Kelvin. El cambio en entalpía (una medida de energía) conforme el gas se caliente de T 2 es la integral de esta ecuación con respecto a T:

$$\triangle h = \int_{T_1}^{T_2} C_p \, dT \tag{2}$$

5.1 Encuentre el cambio en entalpía del oxígeno gaseoso conforme se calienta de 300 K a 1000 K. Los valores de a, b, c y d para el oxígeno son

$$\begin{aligned} a &= 25.48 \\ b &= 1.520x10^{-2} \\ c &= -0.7155x10^{-5} \\ d &= 1.312x10^{-9} \end{aligned} \tag{3}$$

[51]:
```
a, b, c, d, C_p, T, T_1, T_2, h = sp.symbols('a b c d C_p T T_1 T_2 h')
```

[52]:
```
a = 25.480
b = 1.52 * (10**-2)
c = -0.7155 * (10**-5)
d = 1.312 * (10**-9)
T_1 = 300
T_2 = 1000
```

[53]:
```
C_p = a + b*(T_2) + c*(T_2**2) + d*(T_2**3)
C_p
```

[53]:
```
34.836999999999996
```

[55]:
```
h = sp.Integral(C_p, (T, T_1, T_2))
h
```

[55]:
$$\int_{300}^{1000} 34.837 \, dT$$

[57]:
```
h.doit()
```

[57]:
```
24385.9
```