

# Algorítmica y Programación

---

Enero - Mayo 2020

Dr. Iván S. Razo Zapata  
(ivan.razo@itam.mx)

# Numpy

# Numpy



- Linalg
- Modulo para algebra lineal

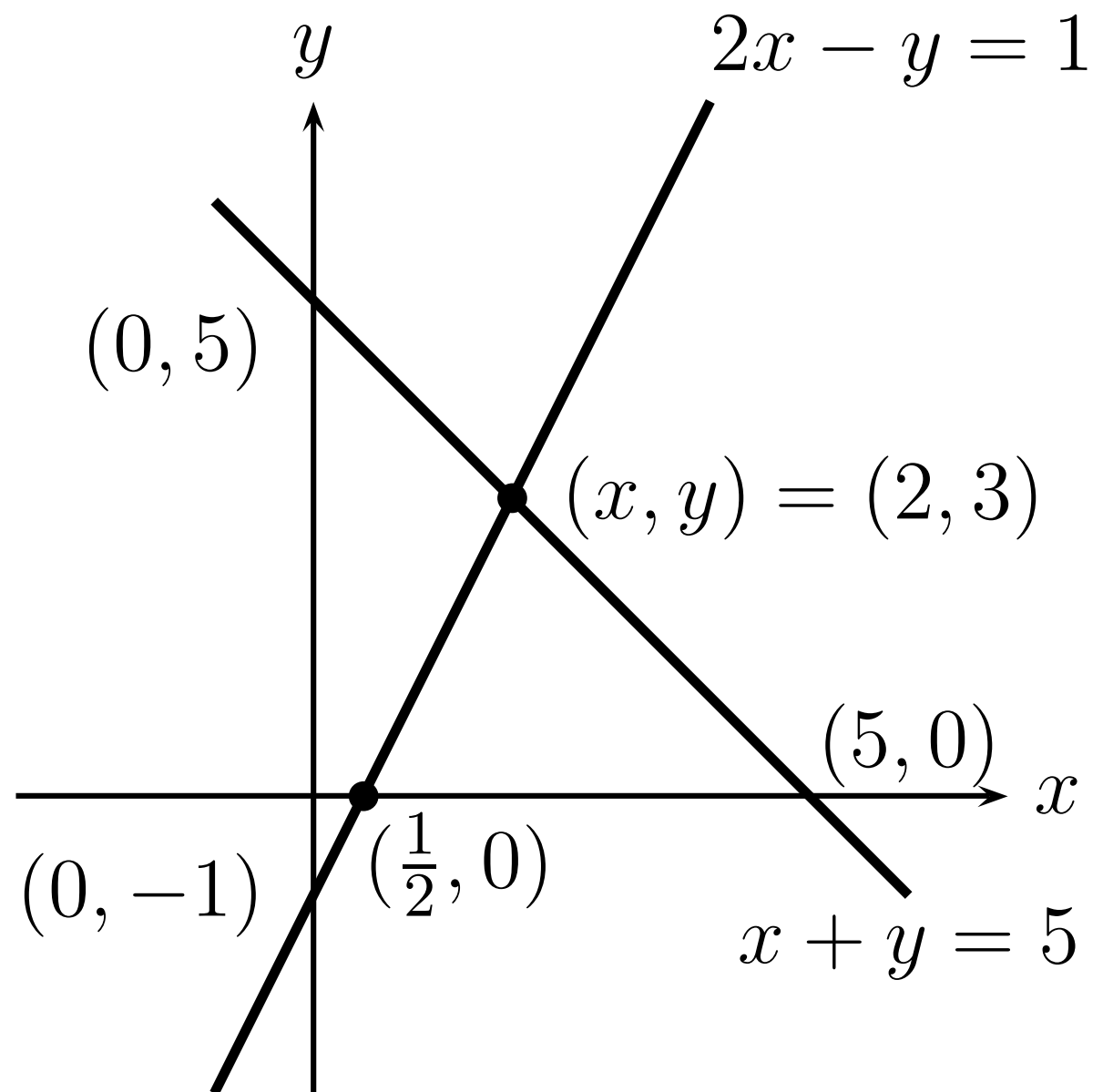
[Scipy.org](#)[Docs](#)[NumPy v1.18 Manual](#)[NumPy Reference](#)[Routines](#)

## Linear algebra (numpy.linalg)

The NumPy linear algebra functions rely on BLAS and LAPACK to provide efficient low level implementations of standard linear algebra algorithms. Those libraries may be provided by NumPy itself using C versions of a subset of their reference implementations but, when possible, highly optimized libraries that take advantage of specialized processor functionality are preferred. Examples of such libraries are [OpenBLAS](#), MKL (TM), and ATLAS. Because those libraries are multithreaded and processor dependent, environmental variables and external packages such as [threadpoolctl](#) may be needed to control the number of threads or specify the processor architecture.

# Sistema de ecuaciones lineales - Filas

$$\begin{array}{rcl} 2x & - & y = 1 \\ x & + & y = 5. \end{array}$$

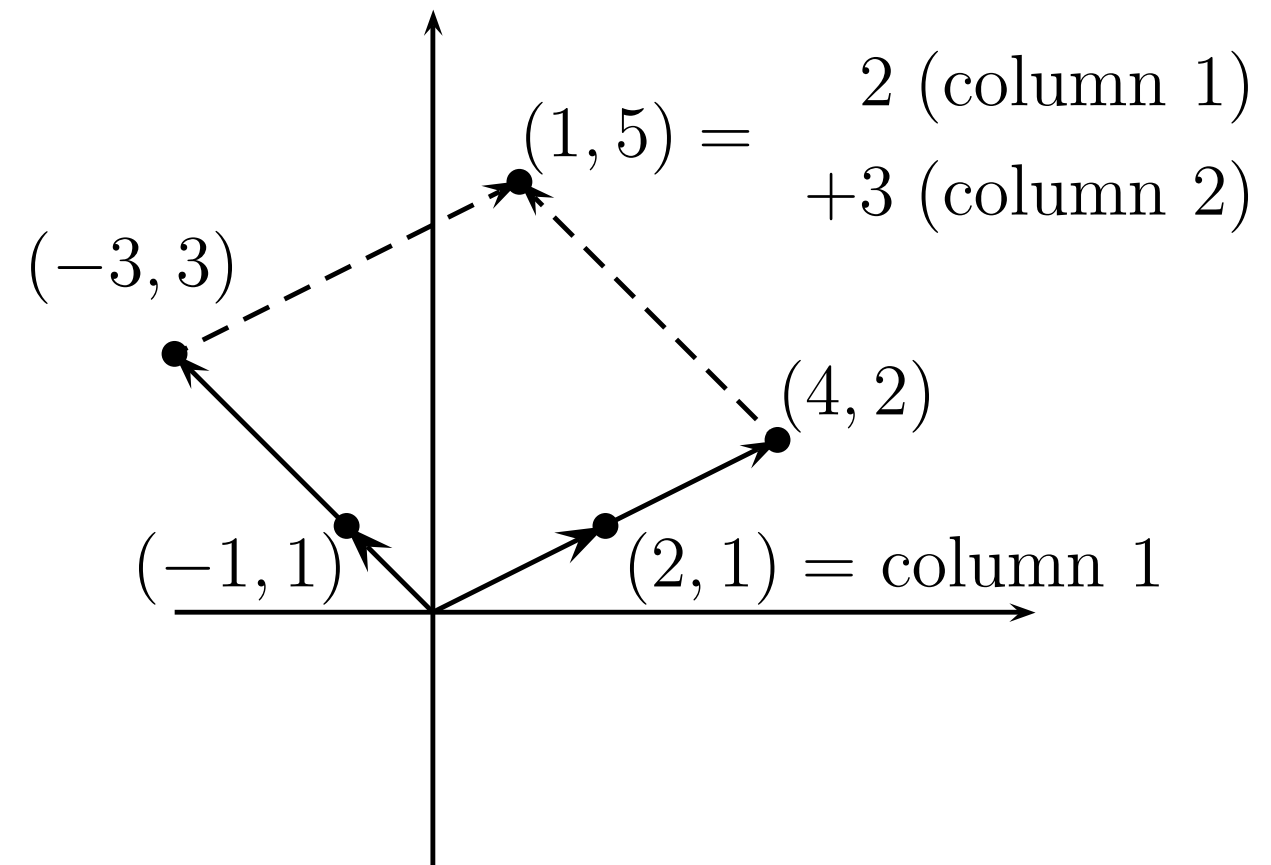


# Sistema de ecuaciones lineales - Columnas

$$\begin{array}{rcl} 2x & - & y = 1 \\ x & + & y = 5. \end{array}$$

Forma vectorial

$$x \begin{bmatrix} 2 \\ 1 \end{bmatrix} + y \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

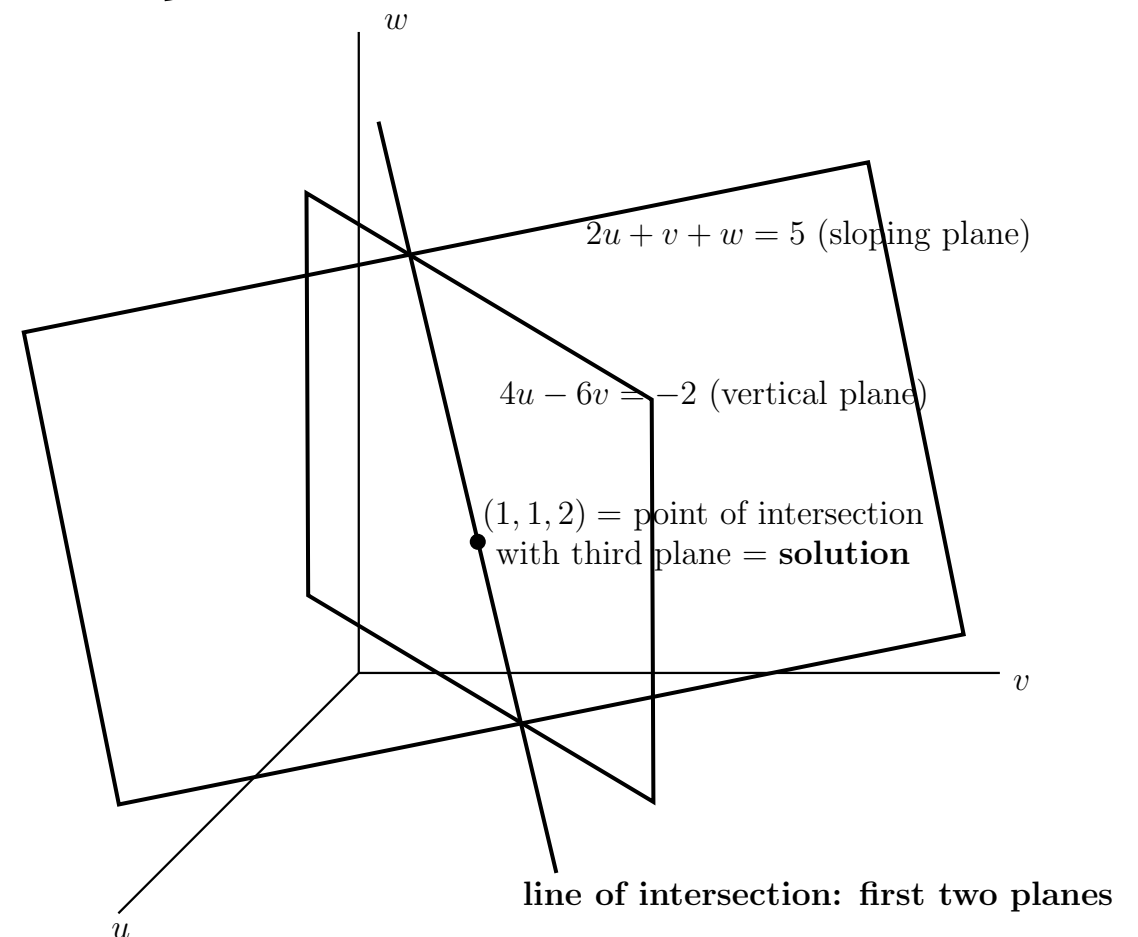


# Sistema de ecuaciones lineales - Filas

$$2u + v + w = 5$$

$$4u - 6v = -2$$

$$-2u + 7v + 2w = 9.$$



# Sistema de ecuaciones lineales - Columnas

---

$$2u + v + w = 5$$

$$4u - 6v = -2$$

$$-2u + 7v + 2w = 9.$$

$$u \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix} + v \begin{bmatrix} 1 \\ -6 \\ 7 \end{bmatrix} + w \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix} = b$$

# Sistema de ecuaciones lineales - Columnas

---

$$2u + v + w = 5$$

$$4u - 6v = -2$$

$$-2u + 7v + 2w = 9.$$

$$\mathbf{1} \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix} + \mathbf{1} \begin{bmatrix} 1 \\ -6 \\ 7 \end{bmatrix} + \mathbf{2} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix}.$$



# Sistema de ecuaciones lineales - Filas

---

$$\begin{array}{rcl} 2x & - & y = 1 \\ x & + & y = 5. \end{array}$$

```
import numpy as np  
  
a = np.array([[2,-1], [1,1]])  
  
b = np.array([1,5])  
  
x = np.linalg.solve(a, b)
```

# Sistema de ecuaciones lineales - Filas

---

$$2u + v + w = 5$$

$$4u - 6v = -2$$

$$-2u + 7v + 2w = 9.$$

```
a2 = np.array([ [2,1,1],  
                [4,-6,0],  
                [-2,7,2] ])
```

```
b2 = np.array([5,-2,9])
```

```
x2 = np.linalg.solve(a2, b2)
```

# Sistema de ecuaciones lineales - Filas

---

```
a2 = np.array([ [2,1,1],  
                [4,-6,0],  
                [-2,7,2] ])
```

```
b2 = np.array([5,-2,9])
```

```
x2 = np.linalg.solve(a2, b2)
```

```
val = np.dot(a2,x2)
```

```
verifica = np.allclose(val, b2)
```

# Ejercicios

---

$$x_1 + 3x_2 + 4x_3 = 3$$

$$2x_1 + 7x_2 + 3x_3 = -7$$

$$2x_1 + 8x_2 + 6x_3 = -4$$

# Ejercicios

---

$$\begin{aligned}x + 2y &= -3 \\ -2x - 4y &= 5\end{aligned}$$

# Ejercicios

---

Ana y Roberto trabajan en la fábrica de una empresa de electrónica y electrodomésticos, Ana está en el área de producción de celulares y Roberto en el área de producción de computadoras.

El día lunes, durante su turno, Ana fabricó 8 celulares y Roberto fabricó 5 computadoras. Por las producciones de ambos, le hicieron ganar a la empresa \$61,000.00.

El día martes Ana fabricó 11 celulares y Roberto fabricó 8 computadoras. Y por las producciones de ambos le hicieron ganar a la empresa la cantidad de \$89,500.00. Ana y Roberto sabían el número de productos que fabricaron cada día y también sabían la cantidad de dinero que las producciones de ambos le hicieron ganar a la empresa. ¿Cuál es el precio de cada uno de los productos que fabrican?

# Ejercicios

---

1. En el triángulo ABC, el ángulo B mide 8 grados menos que la mitad del ángulo A y el ángulo C es 28 grados más grande que el ángulo A. ¿Cuánto miden los ángulos?

$$A + B + C = 180$$

$$0.5 A - B = 8$$

$$-A + C = 28$$

```
a7 = np.array([[1,1,1],  
               [-1,0,1],  
               [0.5,-1,0]])
```

```
b7 = np.array([180,28,8])
```

```
r7 = np.linalg.solve(a7,b7)
```

```
r7  
array([64., 24., 92.])
```