Vectores3

September 29, 2024

1 Universidad Autónoma del Estado de México

2 Centro Universitario UAEM Zumpango

2.1 Ingenierá En Computación

2.2 Graficacion Computacional.

Alumno: Jesus Enrique Lugo Ramirez

Profesor: Hazem alvarez

Fecha: 18 de Septiembre del 2024

2.3 Descripcion: GRAFICACION DE VECTORES.

[1]:
```
import numpy as np # soporte para vectores
import matplotlib.pyplot as plt # graficador
```

2.3.1 *Quiver*

La función plt.quiver en Matplotlib se utiliza para crear gráficos de campos de vectores bidimensionales.

Sintaxis plt.quiver(X, Y, U, V)

X, Y: Coordenadas de los puntos en los que se dibujarán los vectores.

U, V: Componentes del vector (en las direcciones X e Y) en cada punto.

2.3.2 *axvline() y axhline()*

Las funciones plt.axvline() y plt.axhline() en Matplotlib se utilizan para dibujar líneas verticales y horizontales en un gráfico

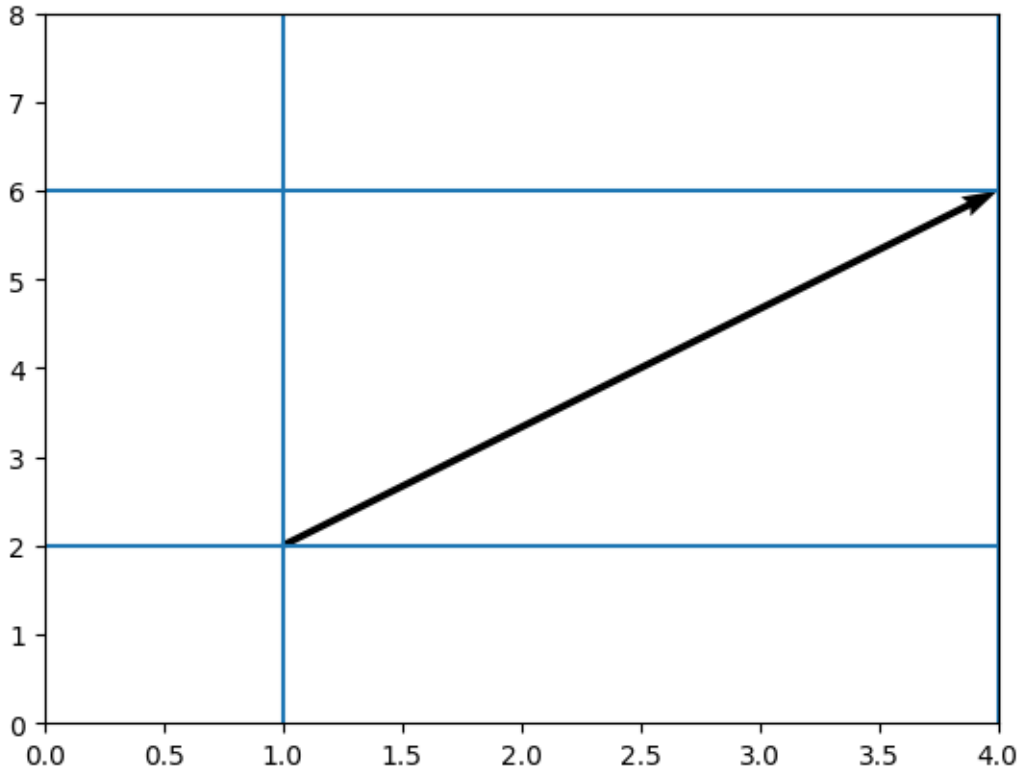
[2]:
```
plt.quiver(1,2, 3,4, scale_units='xy', angles='xy', scale=1)
plt.xlim(0,4)
plt.ylim(0,8)

# guias
plt.axvline(x=1)
```

```
plt.axhline(y=2)

plt.axvline(x=1+3)
plt.axhline(y=2+4)
```

[2]: `<matplotlib.lines.Line2D at 0x119f7bcf9b0>`



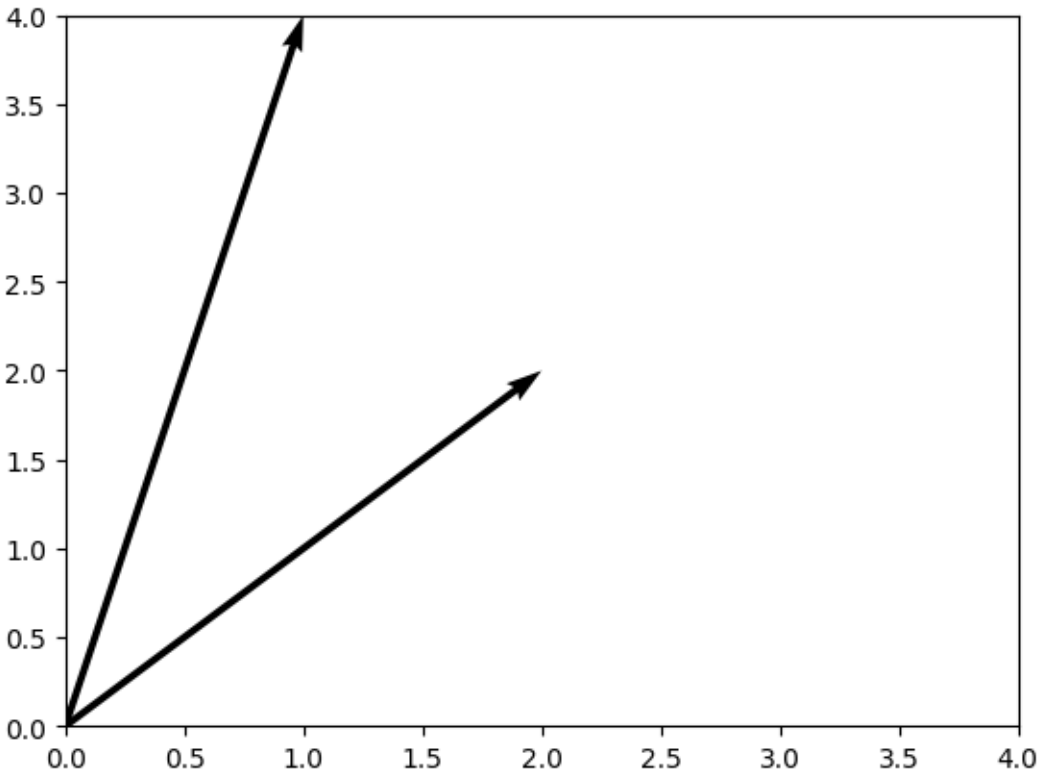
En la grafica anterior se cuenta con un vector que parte desde la coordenada (1,2) y termina hasta la coordenada (4,6).

2.3.3 *xlim() y ylim()*

Las funciones plt.xlim() y plt.ylim() en Matplotlib se utilizan para establecer los límites de los ejes X e Y en un gráfico.

[3]:
```
plt.quiver([0,0],[0,0],[1,2],[4,2], scale_units='xy', angles='xy', scale=1)
plt.xlim(0,4)
plt.ylim(0,4)
```

[3]: `(0.0, 4.0)`



En la gráfica anterior se crean 2 vectores que parten del origen hacia las coordenadas (1,4) y (2,2).

El plano se limita a las medidas 4 en X y 4 en Y

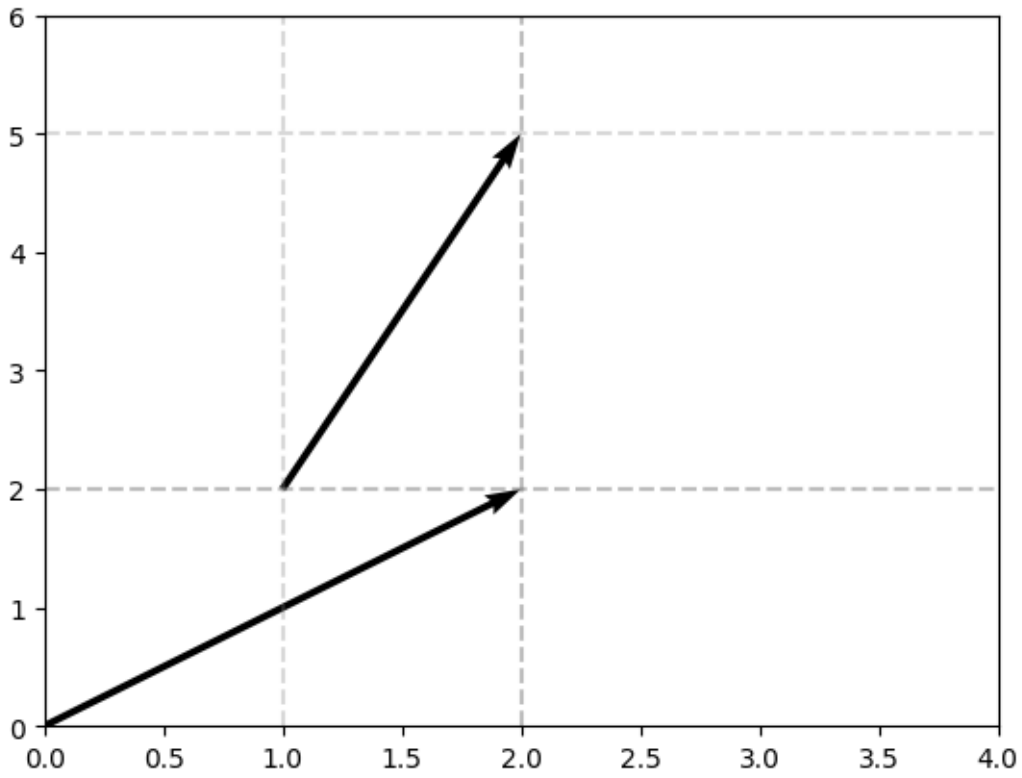
[4]:
```
plt.quiver([1,0],[2,0],[1,2],[3,2], scale_units='xy', angles='xy', scale=1)
plt.xlim(0,4)
plt.ylim(0,6)

plt.axvline(x=1, color='gray', linestyle='--', alpha=0.3)
plt.axhline(y=2, color='gray', linestyle='--', alpha=0.3)

plt.axvline(x=2, color='gray', linestyle='--', alpha=0.3)
plt.axhline(y=2, color='gray', linestyle='--', alpha=0.3)

plt.axvline(x=2, color='gray', linestyle='--', alpha=0.3)
plt.axhline(y=5, color='gray', linestyle='--', alpha=0.3)
```

[4]:
```
<matplotlib.lines.Line2D at 0x119f82479b0>
```



El primer vector parte de las coordenadas (1,2) y tiene componentes (1,3), mientras que el segundo vector comienza en (0,0) y tiene componentes (2,2).

Los parámetros scale_units='xy' y angles='xy' aseguran que las escalas y ángulos de los vectores correspondan al sistema de coordenadas cartesiano.

Luego, se definen los límites del gráfico con plt.xlim(0,4) y plt.ylim(0,6).

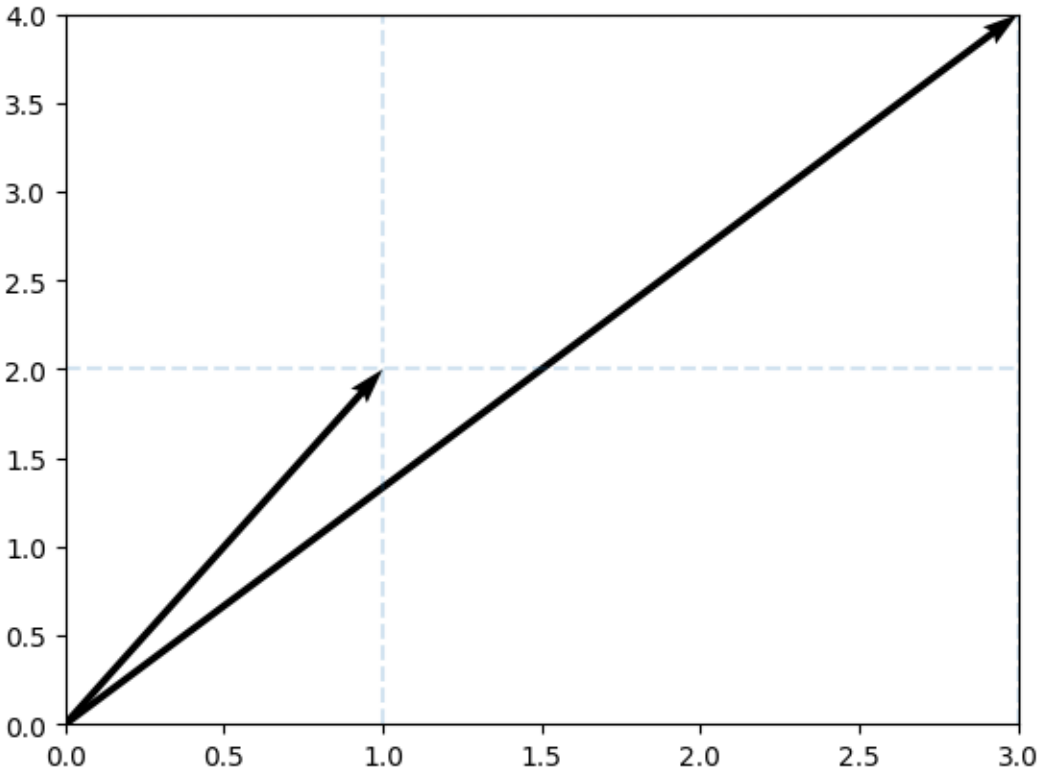
Se añaden líneas punteadas en gris (con baja opacidad) en las posiciones x=1, y=2, x=2, y y=5 utilizando plt.axvline() y plt.axhline().

[5]:
```
v1 = np.array([1,2])
v2 = np.array([3,4])

plt.quiver([0,0],[0,0],[v1[0],v2[0]],[v1[1],v2[1]],angles='xy',␣
 ↪scale_units='xy', scale=1)
plt.xlim(0, max(v1[0], v2[0]))
plt.ylim(0, max(v1[1], v2[1]))

plt.axvline(x=1, alpha=0.2, linestyle='--')
plt.axhline(y=2, alpha=0.2, linestyle='--')
plt.axvline(x=3, alpha=0.2, linestyle='--')
plt.axhline(y=4, alpha=0.2, linestyle='--')
```

[5]: <matplotlib.lines.Line2D at 0x119fa337020>



Este código dibuja dos vectores en un gráfico usando matplotlib. Los vectores v1 = [1, 2] y v2 = [3, 4] parten del origen (0, 0) y terminan en las coordenadas (1, 2) y (3, 4), respectivamente.

Las líneas de referencia (axvline y axhline) se añaden para marcar las posiciones de los extremos de los vectores en los ejes x e y, con un estilo de línea discontinua y algo de transparencia. Los límites del gráfico se ajustan automáticamente para cubrir los extremos de ambos vectores.

plt.axvline() y plt.axhline(): Dibuja líneas verticales y horizontales en el gráfico como referencia, en las posiciones donde terminan los vectores. alpha=0.2 añade transparencia y linestyle='–' las dibuja con líneas discontinuas.

[6]:
```
def grafvecs(vecs, cols):
    plt.figure(figsize=(5,5))
    # vecs: lista de vectores
    for i in range(len(vecs)):
        vec = vecs[i]
        plt.quiver(0,0, vec[0], vec[1], color = cols[i],
                   angles='xy', scale_units='xy', scale=1)
```

2.3.4 *Explicación de la función*

Esta función grafvecs(vecs, cols) está diseñada para graficar una lista de vectores en un plano bidimensional utilizando la función plt.quiver()

La función recibe dos parámetros:

- `vecs: Una lista de vectores, donde cada vector es una lista o tupla de dos elementos que repr`
- `cols: Una lista de colores, donde cada elemento corresponde al color que tendrá cada vector e`

Se crea una figura de 5x5 pulgadas usando plt.figure(). Esto define el tamaño del gráfico.

Aquí se utiliza plt.quiver() para dibujar un vector desde el origen (0,0) hasta las coordenadas dadas por las componentes del vector vec[0] (componente xx) y vec[1] (componente yy).

color=cols[i]: Se asigna el color correspondiente de la lista cols para este vector.

angles='xy': Especifica que las componentes del vector están en el sistema de coordenadas cartesiano.

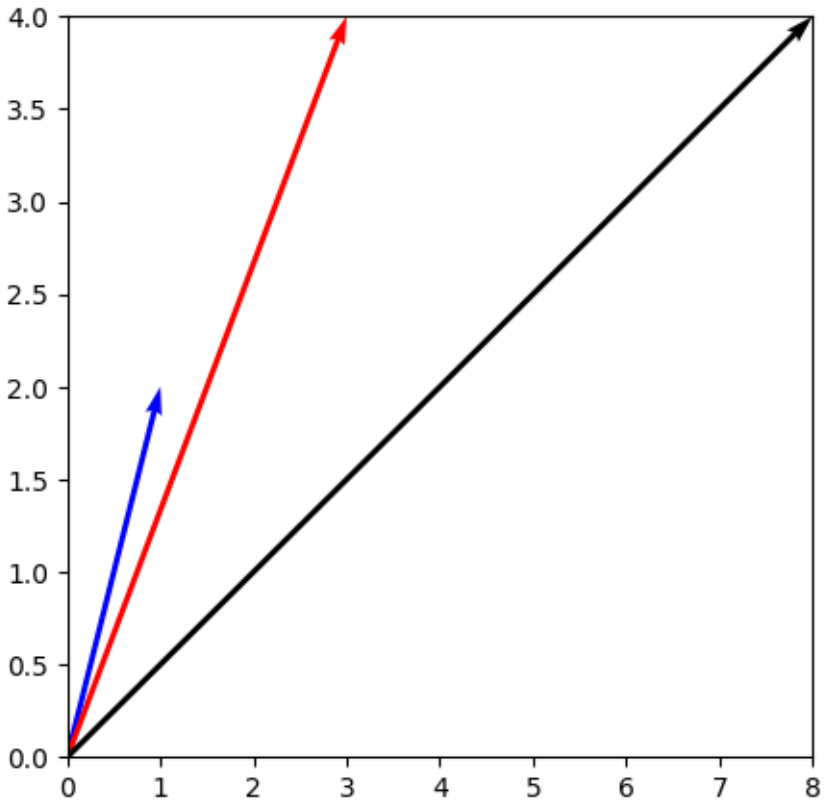
scale_units='xy': Mantiene las escalas de los ejes X e Y consistentes.

scale=1: No se aplica ningún escalado adicional a los vectores (el tamaño de las flechas corresponde exactamente a las componentes del vector).

[7]:
```
v1 = np.array([1,2])
v2 = np.array([3,4])
v3 = np.array([8,4])

grafvecs([v1, v2, v3], ['blue', 'red', 'black'])
plt.xlim(0, max(v1[0], v2[0], v3[0]))
plt.ylim(0, max(v1[1], v2[1], v3[1]))
```

[7]:
```
(0.0, 4.0)
```



Vectores: - v1 = np.array([1,2]): El primer vector tiene las componentes *(1, 2)*. - v2 = np.array([3,4]): El segundo vector tiene las componentes *(3, 4)*. - v3 = np.array([8,4]): El tercer vector tiene las componentes *(8, 4)*.

Configuración de límites: - plt.xlim(0, max(v1[0], v2[0], v3[0])): Establece el límite del eje X desde 0 hasta el valor máximo entre las primeras componentes de los vectores (*1, **3* y *8*), por lo que el eje X va de 0 a 8. - plt.ylim(0, max(v1[1], v2[1], v3[1])): Establece el límite del eje Y desde 0 hasta el valor máximo entre las segundas componentes de los vectores (*2, **4, y** 4*), por lo que el eje Y va de 0 a 4.

Gráfico: La gráfica muestra tres vectores que parten del origen (0,0): - El vector azul va hacia la coordenada (1,2). - El vector rojo se dirige hacia (3,4). - El vector negro se extiende hasta (8,4).