

Graficacion_Vectores_Clase1

September 29, 2024

1 Universidad Autónoma del Estado de México

2 Centro Universitario UAEM Zumpango

2.1 Ingeniería En Computación

2.2 Graficacion Computacional.

Alumno: Jesus Enrique Lugo Ramirez

Profesor: Hazem alvarez

Fecha: 10 de Septiembre del 2024

2.2.1 Descripcion: Grafica los vectores realizados en clase.

```
[1]: #Importando librerias.  
import numpy as np  
import matplotlib.pyplot as plt
```

3 Vector B

```
[2]: # Graficando el vector  
  
x1 = -6 # Valor mínimo para el eje X  
x2 = 6  # Valor máximo para el eje X  
  
y1 = -6 # Valor mínimo para el eje Y  
y2 = 6  # Valor máximo para el eje Y  
  
# Editamos los ejes del gráfico  
plt.axis([x1, x2, y1, y2])  
  
# Activamos la visualización de los ejes  
plt.axis('on')  
  
# Mostramos la cuadrícula en el gráfico  
plt.grid(True)
```

```

# Asignamos un título al gráfico
plt.title("EJERCICIO VECTOR B")

# Definimos la distancia entre puntos en el eje Y
dy = 5

# Definimos la distancia entre puntos en el eje X
dx = 5

# Graficar puntos en el plano
for x in np.arange(x1, x2, dx): # Iteramos sobre el rango de valores para X
    for y in np.arange(y1, y2, dy): # Iteramos sobre el rango de valores para Y
        # Graficamos los puntos con coordenadas (x, y)
        plt.scatter(x, y, s=1.5, color='lightgray') # s es el tamaño del
        ↪ punto, color es el color del punto

# Graficar vectores (flechas)
# plt.arrow(inicio_x, inicio_y, desplazamiento_x, desplazamiento_y,
        ↪ head_length=tamaño de la punta, head_width=ancho de la punta, color=color)

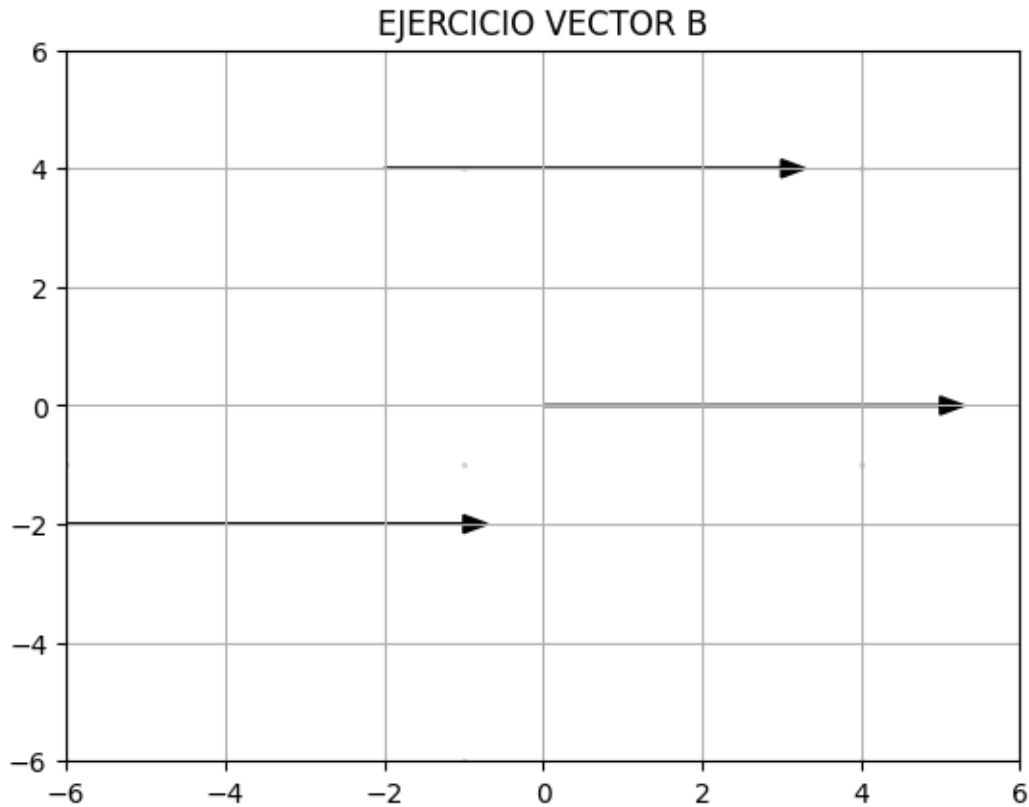
# Primera flecha, desde (0, 0) con desplazamiento de 5 unidades en el eje X
plt.arrow(0, 0, 5, 0, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Segunda flecha, desde (-2, 4) con desplazamiento de 5 unidades en el eje X
plt.arrow(-2, 4, 5, 0, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Tercera flecha, desde (-6, -2) con desplazamiento de 5 unidades en el eje X
plt.arrow(-6, -2, 5, 0, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

```

[2]: <matplotlib.patches.FancyArrow at 0x1ea1477ed50>



4 Vector C

```
[3]: # Graficando el vector

x1 = -6 # Valor mínimo para el eje X
x2 = 6  # Valor máximo para el eje X

y1 = -6 # Valor mínimo para el eje Y
y2 = 6  # Valor máximo para el eje Y

# Editamos los ejes del gráfico
plt.axis([x1, x2, y1, y2])

# Activamos la visualización de los ejes
plt.axis('on')

# Mostramos la cuadrícula en el gráfico
plt.grid(True)

# Asignamos un título al gráfico
```

```

plt.title("EJERCICIO VECTOR C")

# Definimos la distancia entre puntos en el eje Y
dy = 5

# Definimos la distancia entre puntos en el eje X
dx = 5

# Graficar puntos en el plano
for x in np.arange(x1, x2, dx): # Iteramos sobre el rango de valores para X
    for y in np.arange(y1, y2, dy): # Iteramos sobre el rango de valores para Y
        # Graficamos los puntos con coordenadas (x, y)
        plt.scatter(x, y, s=1.5, color='lightgray') # s es el tamaño del
        ↪ punto, color es el color del punto

# Graficar vectores (flechas)
# plt.arrow(inicio_x, inicio_y, desplazamiento_x, desplazamiento_y,
        ↪ head_length=tamaño de la punta, head_width=ancho de la punta, color=color)

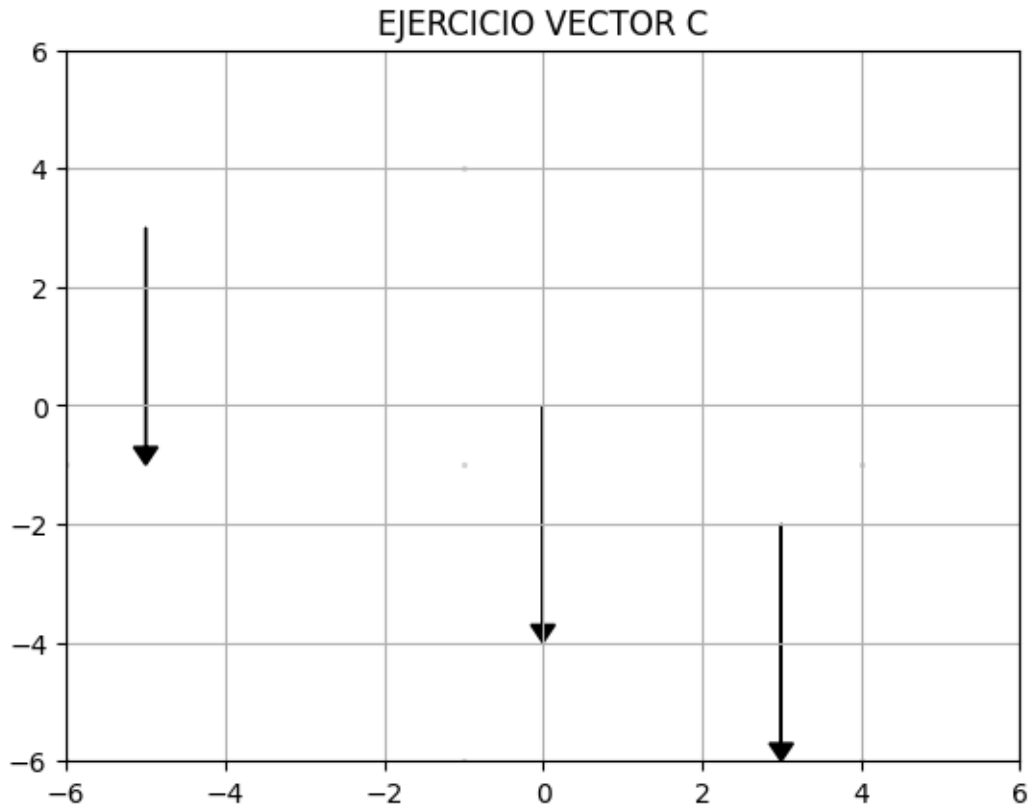
# Primera flecha, desde (0, 0) con desplazamiento hacia abajo en el eje Y (-3.7)
plt.arrow(0, 0, 0, -3.7, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Segunda flecha, desde (-5, 3) con desplazamiento hacia abajo en el eje Y (-3.
        ↪ 7)
plt.arrow(-5, 3, 0, -3.7, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Tercera flecha, desde (3, -2) con desplazamiento hacia abajo en el eje Y (-3.
        ↪ 7)
plt.arrow(3, -2, 0, -3.7, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

```

[3]: <matplotlib.patches.FancyArrow at 0x1ea147c15b0>



5 Vector D

```
[4]: # Graficando el vector

x1 = -6 # Valor mínimo para el eje X
x2 = 6  # Valor máximo para el eje X

y1 = -6 # Valor mínimo para el eje Y
y2 = 6  # Valor máximo para el eje Y

# Editamos los ejes del gráfico
plt.axis([x1, x2, y1, y2])

# Activamos la visualización de los ejes
plt.axis('on')

# Mostramos la cuadrícula en el gráfico
plt.grid(True)

# Asignamos un título al gráfico
```

```

plt.title("EJERCICIO VECTOR D")

# Definimos la distancia entre puntos en el eje Y
dy = 5

# Definimos la distancia entre puntos en el eje X
dx = 5

# Graficar puntos en el plano
for x in np.arange(x1, x2, dx): # Iteramos sobre el rango de valores para X
    for y in np.arange(y1, y2, dy): # Iteramos sobre el rango de valores para Y
        # Graficamos los puntos con coordenadas (x, y)
        plt.scatter(x, y, s=1.5, color='lightgray') # s es el tamaño del
        ↪ punto, color es el color del punto

# Graficar vectores (flechas)
# plt.arrow(inicio_x, inicio_y, desplazamiento_x, desplazamiento_y,
        ↪ head_length=tamaño de la punta, head_width=ancho de la punta, color=color)

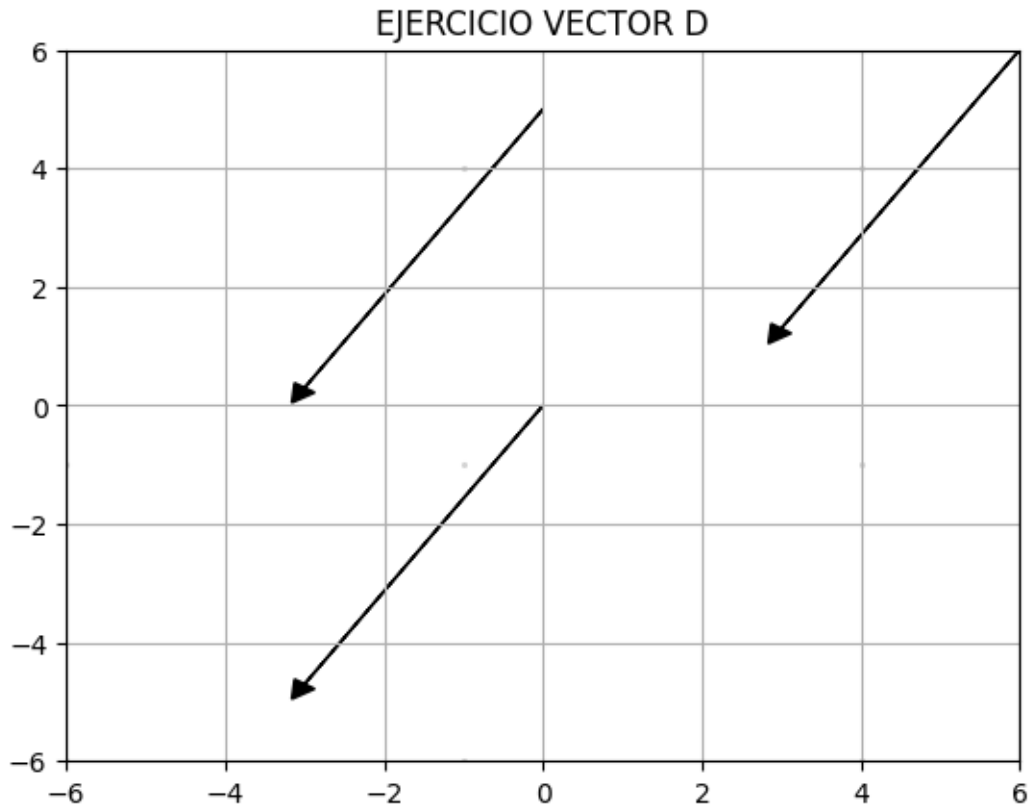
# Primera flecha, desde (0, 0) con desplazamiento de (-3, -4.7)
plt.arrow(0, 0, -3, -4.7, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Segunda flecha, desde (0, 5) con desplazamiento de (-3, -4.7)
plt.arrow(0, 5, -3, -4.7, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Tercera flecha, desde (6, 6) con desplazamiento de (-3, -4.7)
plt.arrow(6, 6, -3, -4.7, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

```

[4]: <matplotlib.patches.FancyArrow at 0x1ea16a9f380>



6 Vector MN

```
[5]: # Graficando el vector

x1 = -6 # Valor mínimo para el eje X
x2 = 6  # Valor máximo para el eje X

y1 = -6 # Valor mínimo para el eje Y
y2 = 6  # Valor máximo para el eje Y

# Editamos los ejes del gráfico
plt.axis([x1, x2, y1, y2])

# Activamos la visualización de los ejes
plt.axis('on')

# Mostramos la cuadrícula en el gráfico
plt.grid(True)

# Asignamos un título al gráfico
```

```

plt.title("EJERCICIO VECTOR MN")

# Definimos la distancia entre puntos en el eje Y
dy = 5

# Definimos la distancia entre puntos en el eje X
dx = 5

# Graficar puntos en el plano
for x in np.arange(x1, x2, dx): # Iteramos sobre el rango de valores para X
    for y in np.arange(y1, y2, dy): # Iteramos sobre el rango de valores para Y
        # Graficamos los puntos con coordenadas (x, y)
        plt.scatter(x, y, s=1.5, color='lightgray') # s es el tamaño del
        ↪ punto, color es el color del punto

# Graficar vectores (flechas)
# plt.arrow(inicio_x, inicio_y, desplazamiento_x, desplazamiento_y,
        ↪ head_length=tamaño de la punta, head_width=ancho de la punta, color=color)

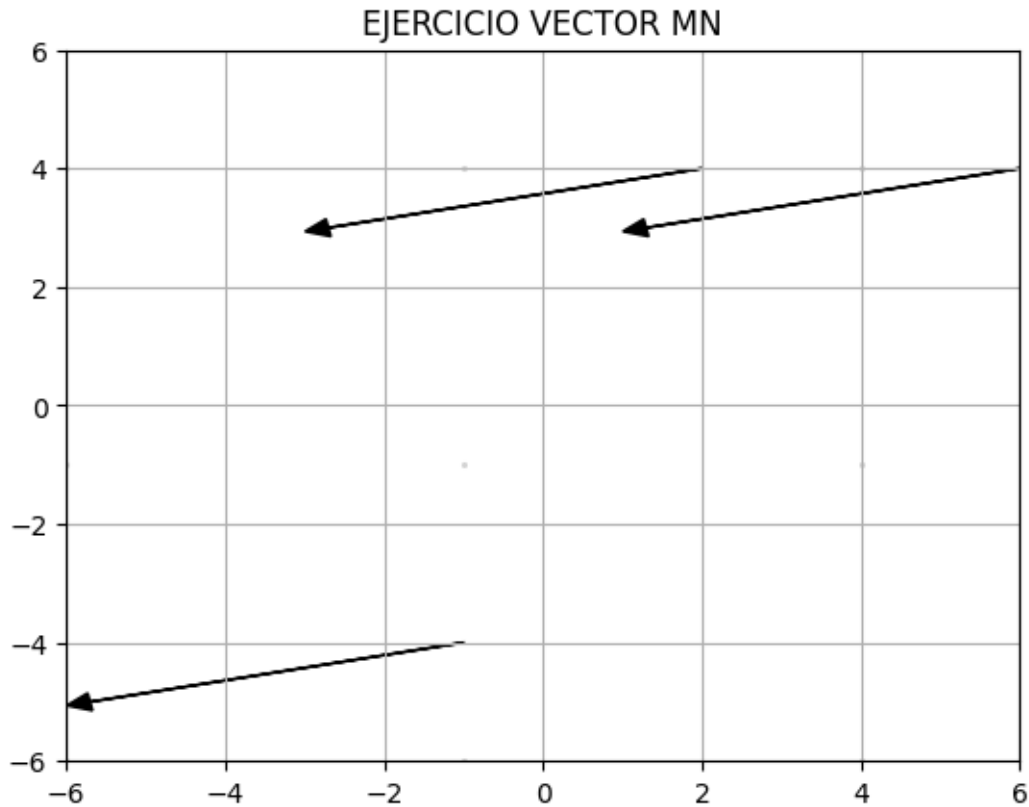
# Primera flecha, desde (2, 4) con desplazamiento de (-4.7, -1)
plt.arrow(2, 4, -4.7, -1, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Segunda flecha, desde (6, 4) con desplazamiento de (-4.7, -1)
plt.arrow(6, 4, -4.7, -1, head_length=0.3, head_width=0.3, color="k") # Flecha
        ↪ negra

# Tercera flecha, desde (-1, -4) con desplazamiento de (-4.7, -1)
plt.arrow(-1, -4, -4.7, -1, head_length=0.3, head_width=0.3, color="k") #
        ↪ Flecha negra

```

[5]: <matplotlib.patches.FancyArrow at 0x1ea16b43ef0>



7 Vector PN

```
[6]: # Graficando el vector

x1 = -6 # Valor mínimo para el eje X
x2 = 6  # Valor máximo para el eje X

y1 = -6 # Valor mínimo para el eje Y
y2 = 6  # Valor máximo para el eje Y

# Editamos los ejes del gráfico
plt.axis([x1, x2, y1, y2])

# Activamos la visualización de los ejes
plt.axis('on')

# Mostramos la cuadrícula en el gráfico
plt.grid(True)

# Asignamos un título al gráfico
```

```

plt.title("EJERCICIO VECTOR PN")

# Definimos la distancia entre puntos en el eje Y
dy = 5

# Definimos la distancia entre puntos en el eje X
dx = 5

# Graficar puntos en el plano
for x in np.arange(x1, x2, dx): # Iteramos sobre el rango de valores para X
    for y in np.arange(y1, y2, dy): # Iteramos sobre el rango de valores para Y
        # Graficamos los puntos con coordenadas (x, y)
        plt.scatter(x, y, s=1.5, color='lightgray') # s es el tamaño del
        ↪ punto, color es el color del punto

# Graficar vectores (flechas)
# plt.arrow(inicio_x, inicio_y, desplazamiento_x, desplazamiento_y,
        ↪ head_length=tamaño de la punta, head_width=ancho de la punta, color=color)

# Primera flecha, desde (6, -1) con desplazamiento de (-3.7, 5.7)
plt.arrow(6, -1, -3.7, 5.7, head_length=0.3, head_width=0.3, color="k") #
        ↪ Flecha negra

# Segunda flecha, desde (1, -4) con desplazamiento de (-3.7, 5.7)
plt.arrow(1, -4, -3.7, 5.7, head_length=0.3, head_width=0.3, color="k") #
        ↪ Flecha negra

# Tercera flecha, desde (-2, 0) con desplazamiento de (-3.7, 5.7)
plt.arrow(-2, 0, -3.7, 5.7, head_length=0.3, head_width=0.3, color="k") #
        ↪ Flecha negra

```

[6]: <matplotlib.patches.FancyArrow at 0x1ea147a6f90>

