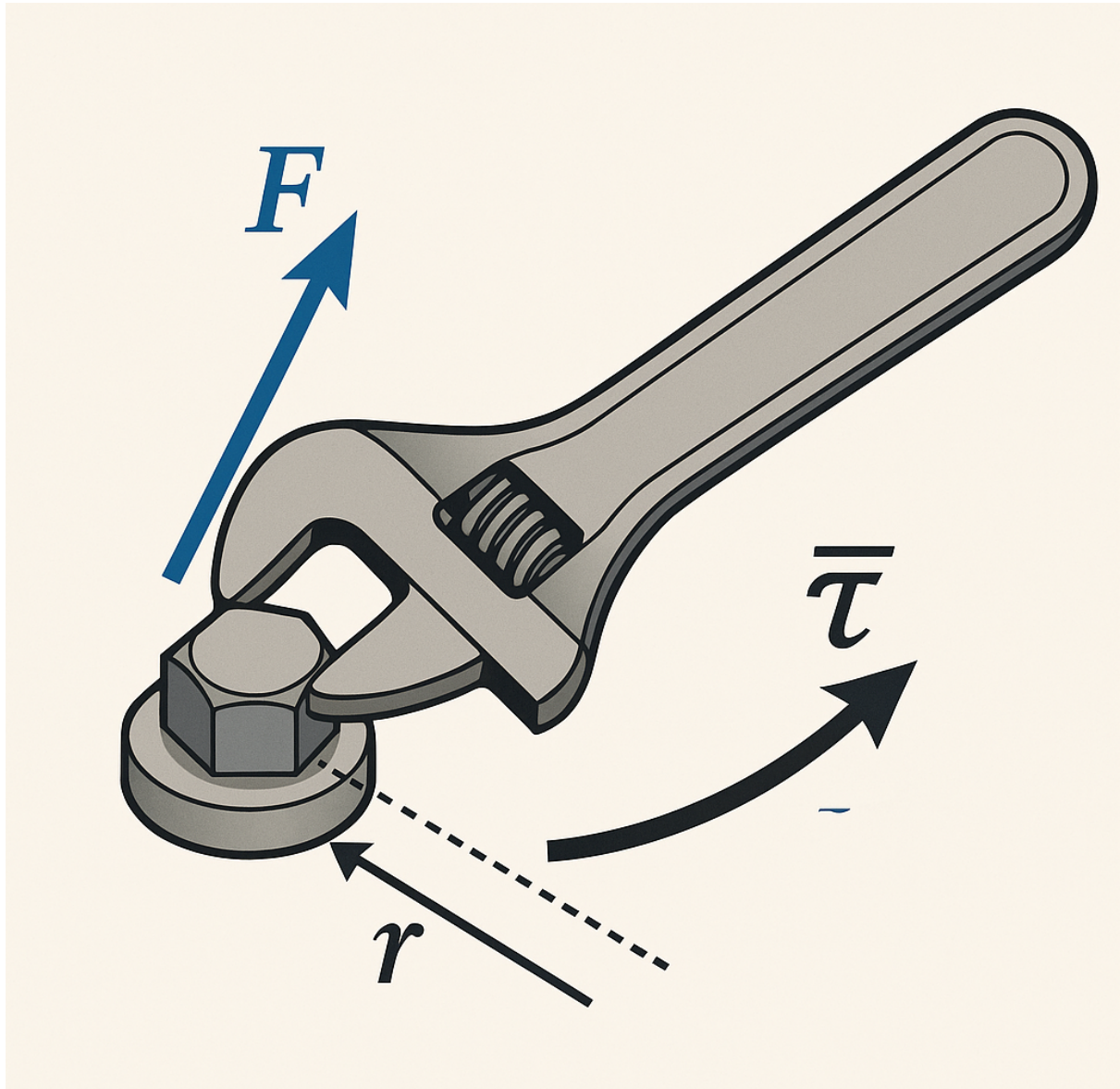


Informe Propuesto sobre Fuerza de Rotación (Torque)

Título:

Cálculo del Torque Aplicado sobre una Llave Inglesa utilizando Python



Autor:

Juan Ignacio Suhurt Fernández

Fecha:

julio 2025

Resumen:

Este trabajo explora el concepto de torque o momento de fuerza, aplicado al caso de una llave inglesa que gira una tuerca. Se desarrolla el marco conceptual de la fuerza de rotación, y se aplica un programa en Python para calcular el torque en función de la fuerza aplicada y la distancia al eje de giro. El trabajo incluye gráficos que muestran cómo varía el torque con diferentes parámetros. Esta aproximación computacional permite modelar situaciones físicas y verificar resultados teóricos. El trabajo puede ser una herramienta didáctica útil en la enseñanza de la dinámica rotacional.

Palabras clave:

Torque, Momento de Fuerza, Python, Dinámica Rotacional

Introducción:

El torque es una de las magnitudes fundamentales en la física del movimiento rotacional. Su importancia va desde sistemas simples, como palancas o herramientas manuales, hasta sistemas complejos como motores y estructuras en ingeniería. A diferencia de la fuerza lineal, el torque tiene un punto de aplicación respecto a un eje, y su intensidad depende no solo de la magnitud de la fuerza sino también de la distancia al punto de giro.

Este trabajo se enfoca en aplicar los conocimientos teóricos del torque a un ejemplo cotidiano y fácilmente comprensible: el uso de una llave inglesa. Para ello se implementa un pequeño programa en Python que permite visualizar cómo el torque cambia con diferentes parámetros. Este tipo de herramientas computacionales permite enriquecer la enseñanza de la física, facilitando el análisis de fenómenos complejos.

Desarrollo:

El torque o momento de una fuerza (llamémoslo τ) es la capacidad de una fuerza de producir una rotación respecto a un eje que pase por el punto O (centro de momentos)

$$\tau = r \times F$$

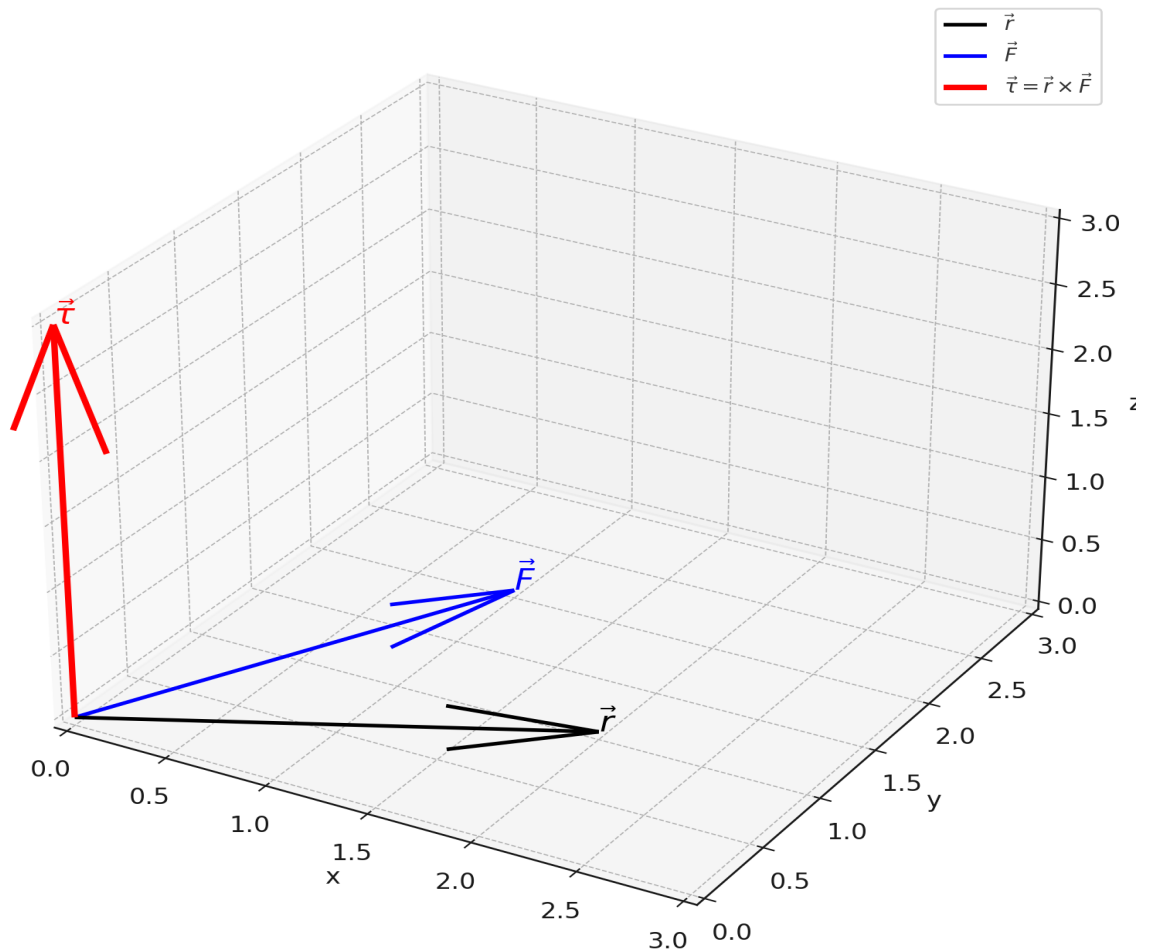
Donde r es la posición del punto de aplicación de la fuerza respecto de O y F la magnitud de la fuerza ejercida en dicho punto.

$$[\tau] = \text{Nm} \quad \tau = r \times F \times \sin(\theta) \text{ (módulo)}$$

donde θ es el ángulo entre la fuerza y la palanca.

τ es perpendicular al plano formado por F y r

Torque como producto vectorial: $\vec{\tau} = \vec{r} \times \vec{F}$



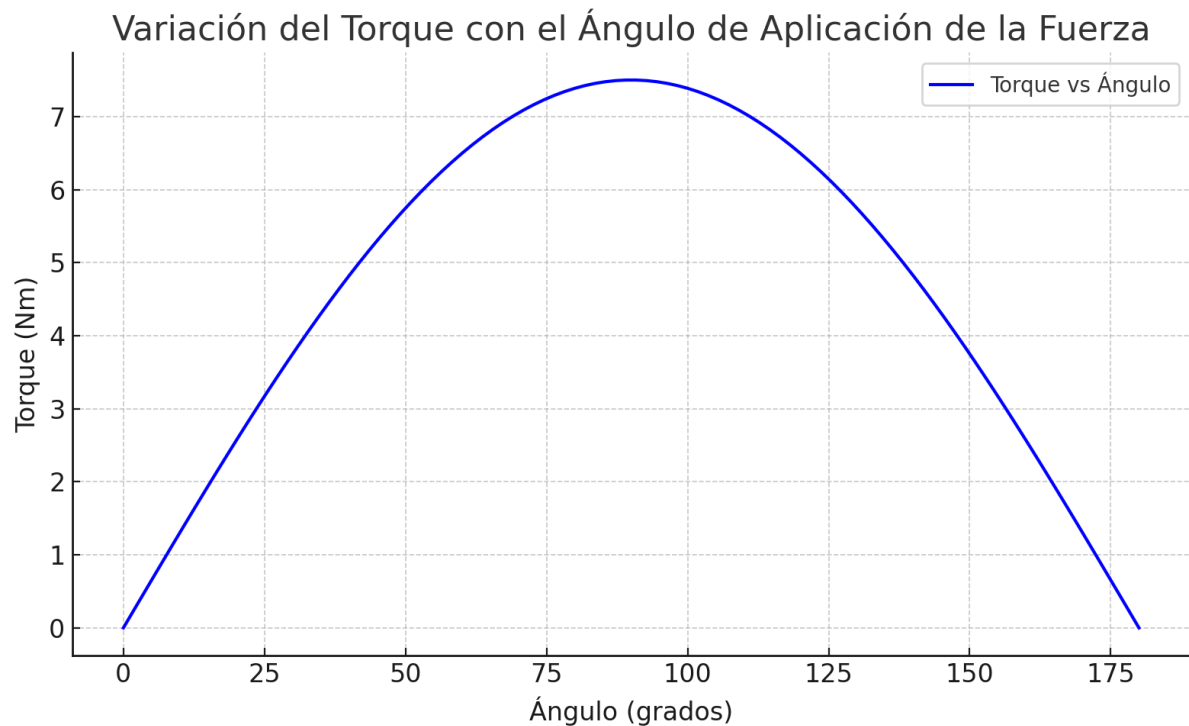
Ejemplo: Torque aplicado sobre una llave inglesa

Imaginemos que aplicamos una fuerza de $F=30\text{ N}$ a una distancia de $r=0.25\text{ m}$ respecto al eje. Si la fuerza es perpendicular ($\theta=90^\circ$), el torque es máximo:

$$\tau = r \times F \times \sin(\theta)$$

$$\tau = 0.25 \times 30 \times \sin(90)$$

$$\tau = 7.5\text{ Nm}$$



Resultados Obtenidos

- El torque máximo ocurre cuando el ángulo entre la fuerza y el brazo de la palanca es de 90° .
- A 0° o 180° , el torque es cero, ya que la fuerza actúa alineada con el brazo de la palanca.
- El gráfico generado permite visualizar esta dependencia de forma clara y útil para fines didácticos.

Programa de torque en función del ángulo

```
import numpy as np
```

```

import matplotlib.pyplot as plt

def calcular_torque(r, F, tita_deg):
    tita_rad = np.radians(tita_deg)
    return r * F * np.sin(tita_rad)

# Parámetros
r = 0.25 # metros
F = 30 # Newtons
ángulos = np.linspace(0, 180, 100)
torques = calcular_torque(r, F, ángulos)

# Gráfico
plt.figure(figsize=(8,5))
plt.plot(ángulos, torques, label='Torque vs Ángulo', color='blue')
plt.title('Variación del Torque con el Ángulo de Aplicación de la Fuerza')
plt.xlabel('Ángulo (grados)')
plt.ylabel('Torque (Nm)')
plt.grid(True)
plt.legend()
plt.savefig("grafico_torque.png")
plt.show()

```

Gráfico insertado en el desarrollo del tema:

```

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Definir los vectores
r = np.array([2, 1, 0]) # Vector de posición
F = np.array([1, 2, 0]) # Vector de fuerza
tau = np.cross(r, F) # Torque = r x F

# Crear figura
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')

# Dibujar los vectores
ax.quiver(0, 0, 0, r[0], r[1], r[2], color='black', label=r'$\vec{r}$', linewidth=2)
ax.quiver(0, 0, 0, F[0], F[1], F[2], color='blue', label=r'$\vec{F}$', linewidth=2)
ax.quiver(0, 0, 0, tau[0], tau[1], tau[2], color='red', label=r'$\vec{\tau} = \vec{r} \times \vec{F}$', linewidth=3)

# Etiquetas grandes en el extremo de cada vector
ax.text(r[0], r[1], r[2], r'$\vec{r}$', color='black', fontsize=16)

```

```

ax.text(F[0], F[1], F[2], r'$\vec{F}$', color='blue', fontsize=16)
ax.text(tau[0], tau[1], tau[2], r'$\vec{\tau}$', color='red', fontsize=16)

# Configurar límites de los ejes
ax.set_xlim([0, 3])
ax.set_ylim([0, 3])
ax.set_zlim([0, 3])

# Etiquetas de ejes
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

# Título del gráfico
ax.set_title(r'Torque como producto vectorial:  $\vec{\tau} = \vec{r} \times \vec{F}$ ',
fontsize=14)

# Leyenda
ax.legend()

# Mostrar gráfico
plt.tight_layout()
plt.show()

```

Programa de torque:

```

import numpy as np
import matplotlib.pyplot as plt

def calcular_torque(r, F, tita_deg):
    """Calcula el torque dado r, F y el ángulo (en grados)."""
    tita_rad = np.radians(tita_deg)
    return r * F * np.sin(tita_rad)

def graficar_torque(r, F, tita_especifico, torque_especifico):
    """Genera un gráfico del torque en función del ángulo."""
    angulos = np.linspace(0, 180, 200)
    torques = calcular_torque(r, F, angulos)

    plt.figure(figsize=(9, 5))
    plt.plot(angulos, torques, label='Torque vs Ángulo', color='navy', linewidth=2)
    plt.plot(tita_especifico, torque_especifico, 'ro', label=f'Torque en {tita_especifico}°')

    plt.title('Variación del Torque con el Ángulo', fontsize=14)
    plt.xlabel('Ángulo (grados)', fontsize=12)
    plt.ylabel('Torque (Nm)', fontsize=12)
    plt.grid(True, linestyle='--', alpha=0.6)

```

```

plt.legend()
plt.tight_layout()
plt.show()

def main():
    print(" Calculadora de Torque con Gráfico\n")

    try:
        r = float(input(" Ingrese la distancia desde el eje (en metros): "))
        F = float(input(" Ingrese la fuerza aplicada (en Newtons): "))
        tita = float(input(" Ingrese el ángulo de aplicación (en grados): "))

        torque = calcular_torque(r, F, tita)
        print(f"\n El torque aplicado a {tita}° es: {torque:.2f} Nm")

        graficar_torque(r, F, tita, torque)

    except ValueError:
        print("\n Error: asegúrese de ingresar solo valores numéricos.")

if __name__ == "__main__":
    main()

```

Referencias:

1. Serway, Raymond A. y Jewett, John W. *Física para Ciencias e Ingeniería*, Volumen 1, 9ª edición, Cengage Learning, 2014. Capítulo sobre Dinámica de Rotación.
2. Material docente UNLPam - Informática
3. Física II – Profesorado y Licenciatura en Física – FCEyN – UNLPam Dinámica de la rotación. Sistemas extensos

