

# ¿Cómo manipular funciones?

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt

#Configurando Latex
# Configuración de Matplotlib para usar LaTeX
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern Roman"],
    "text.latex.preamble": r"\usepackage{amsmath}"
})
```

## Desplazamientos

Siendo  $c$  una constante mayor que cero, entonces la gráfica:

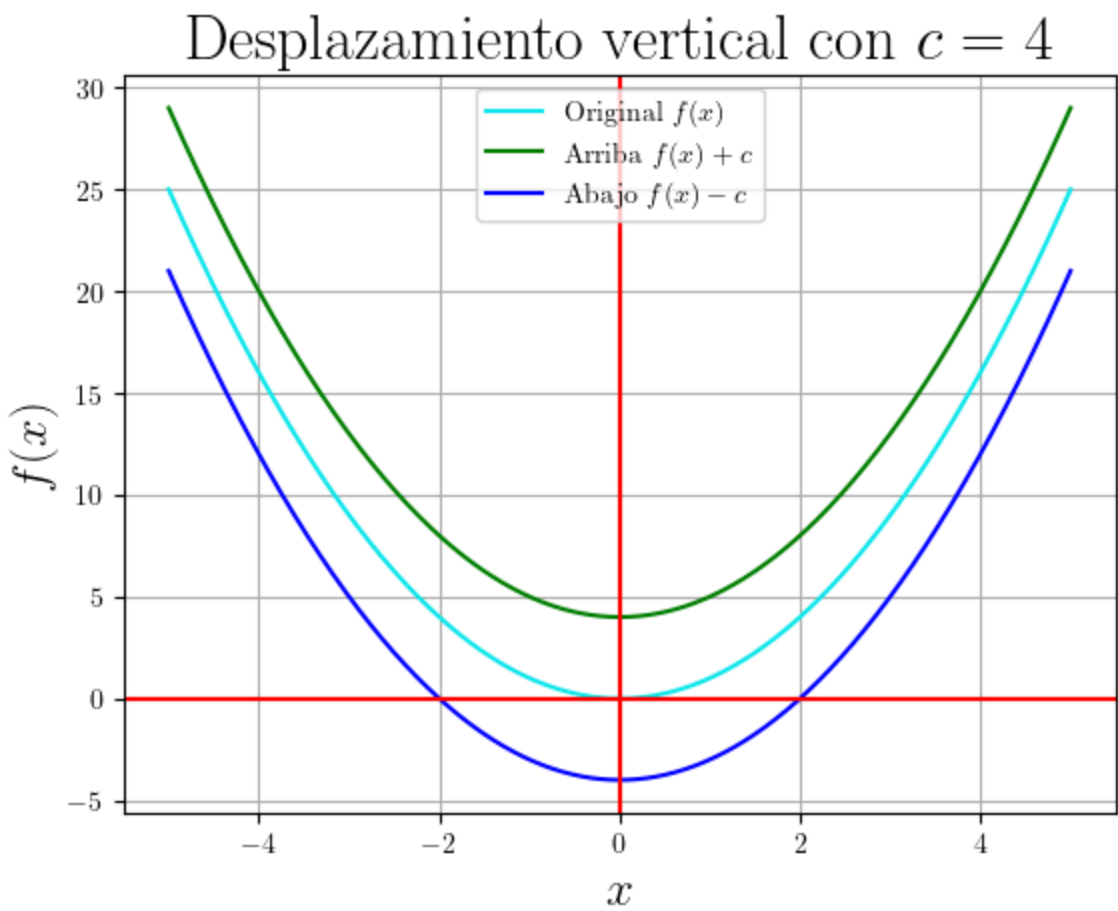
### Vertical

- $y = f(x) + c$  se desplaza  $c$  unidades hacia arriba.
- $y = f(x) - c$  se desplaza  $c$  unidades hacia abajo.

```
In [ ]: # definiendo funcion parabola
def f(x):
    return x**2

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-5,5, num=N)

#Creando graficas
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original  $f(x)$ ',color='#0EE9ED')
ax.plot(x,f(x)+c,label=r'Arriba  $f(x)+c$ ',color='green')
ax.plot(x,f(x)-c,label=r'Abajo  $f(x)-c$ ',color='blue')
ax.grid()
ax.axhline(y=0, color='r')
ax.axvline(x=0, color='r')
plt.title(r'Desplazamiento vertical con  $c=4$ ',fontsize=23)
plt.xlabel(r' $x$ ',fontsize=18)
plt.ylabel(r' $f(x)$ ',fontsize=18)
plt.legend()
plt.show()
```



### Horizontal

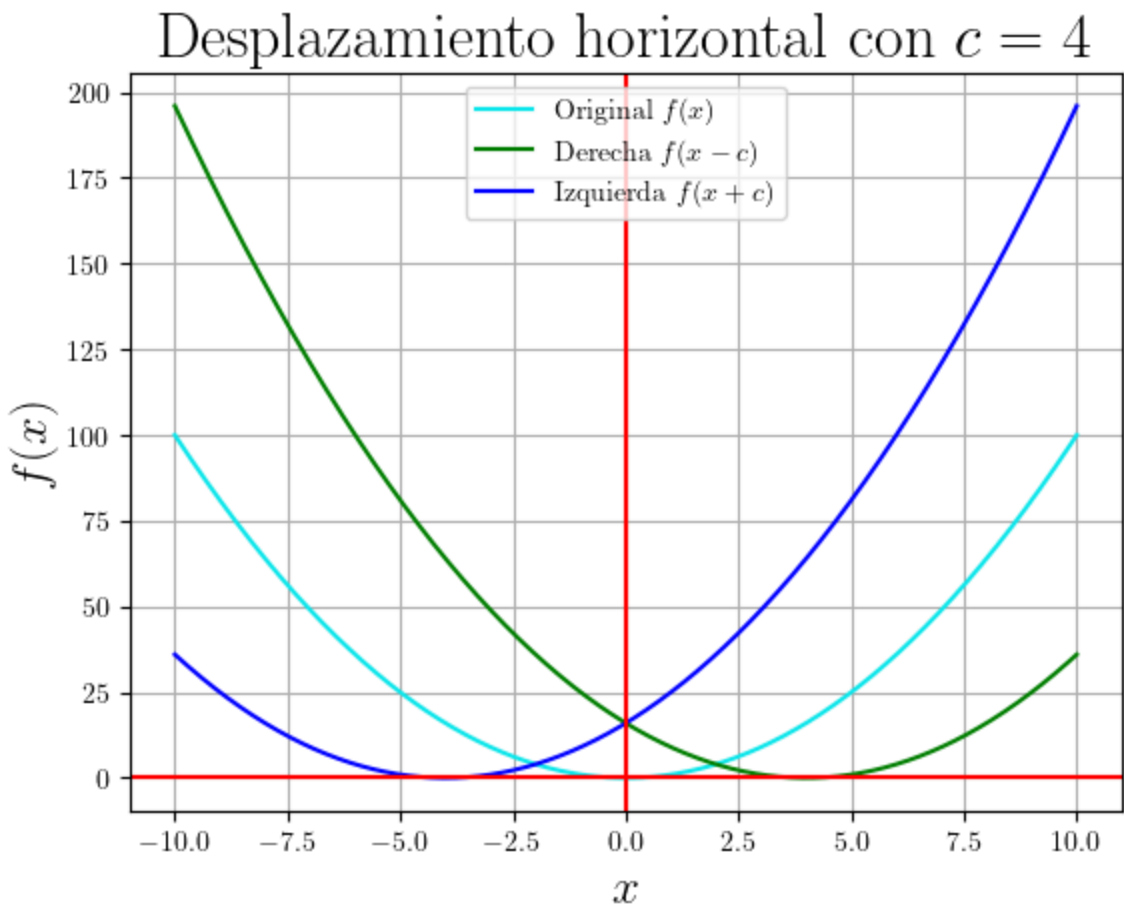
- $y = f(x - c)$  se desplaza  $c$  unidades hacia la derecha.
- $y = f(x + c)$  se desplaza  $c$  unidades hacia la izquierda.

```
In [ ]: # definiendo funcion parabola
def f(x):
```

```
return x**2

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-10,10, num=N)

#Creando graficas
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original $f(x)$',color='#0EE9ED')
ax.plot(x,f(x-c),label=r'Derecha $f(x-c)$',color='green')
ax.plot(x,f(x+c),label=r'Izquierda $f(x+c)$',color='blue')
ax.grid()
ax.axhline(y=0, color='r')
ax.axvline(x=0, color='r')
plt.title(r'Desplazamiento horizontal con $c=4$',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



## Alargamientos y compresiones

Siendo  $c$  una constante mayor que cero, entonces la grfica:

NOTA:

- **Alargamiento:** nos referimos a que el valor incrementa ms rpido, segn en el eje  $x$  o  $y$  planteado.
- **Compresin:** nos referimos a que el valor incrementa ms lento, segn en el eje  $x$  o  $y$  planteado.

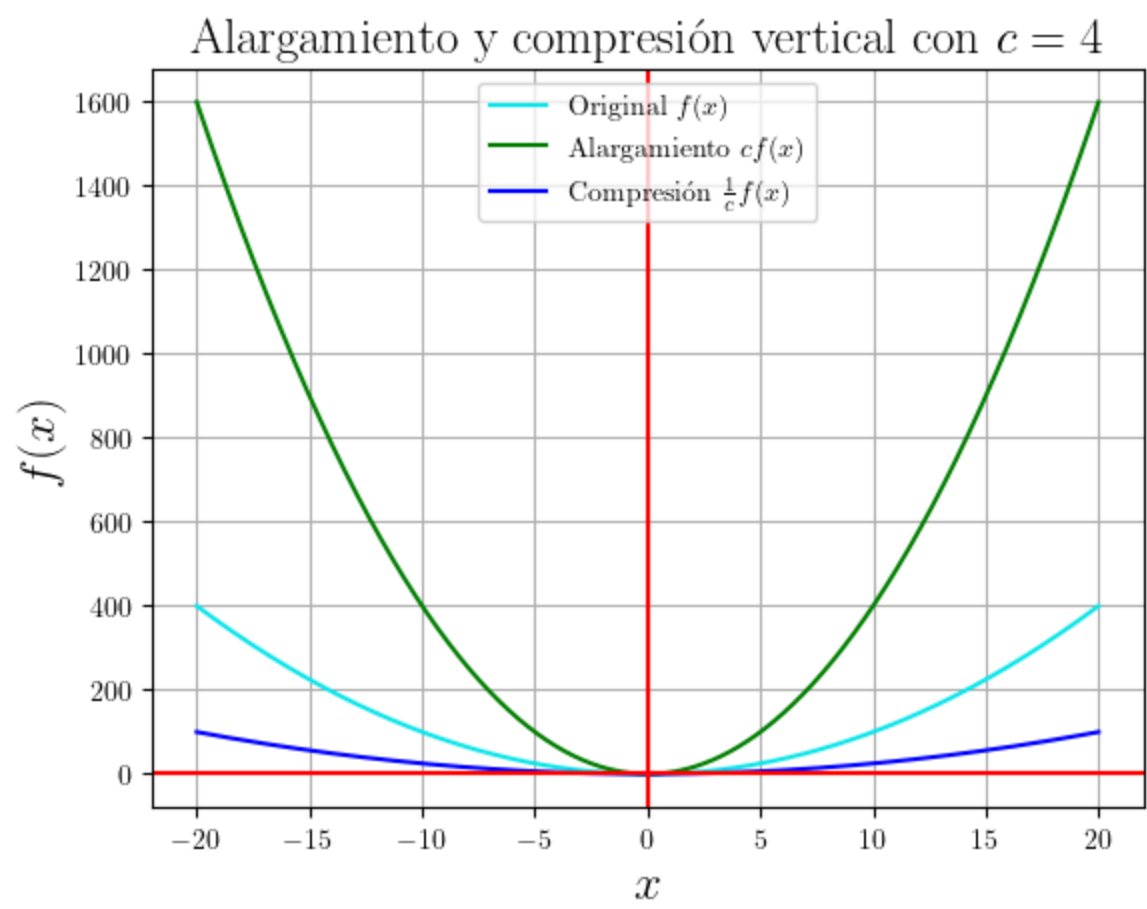
### Vertical

- $y = c \cdot f(x)$  alarga la grfica verticalmente en un factor de  $c$ .
- $y = \frac{1}{c} \cdot f(x)$  comprime la grfica verticalmente en un factor de  $c$ .

```
In [ ]: # definiendo funcion parabola
def f(x):
    return x**2

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-20,20, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original $f(x)$',color='#0EE9ED')
ax.plot(x,c*f(x),label=r'Alargamiento $cf(x)$',color='green')
ax.plot(x,(1/c)*f(x),label=r'Compresin $\frac{1}{c}$ $f(x)$',color='blue')
ax.grid()
ax.axhline(y=0, color='r')
ax.axvline(x=0, color='r')
plt.title(r'Alargamiento y compresin vertical con $c=4$',fontsize=18)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



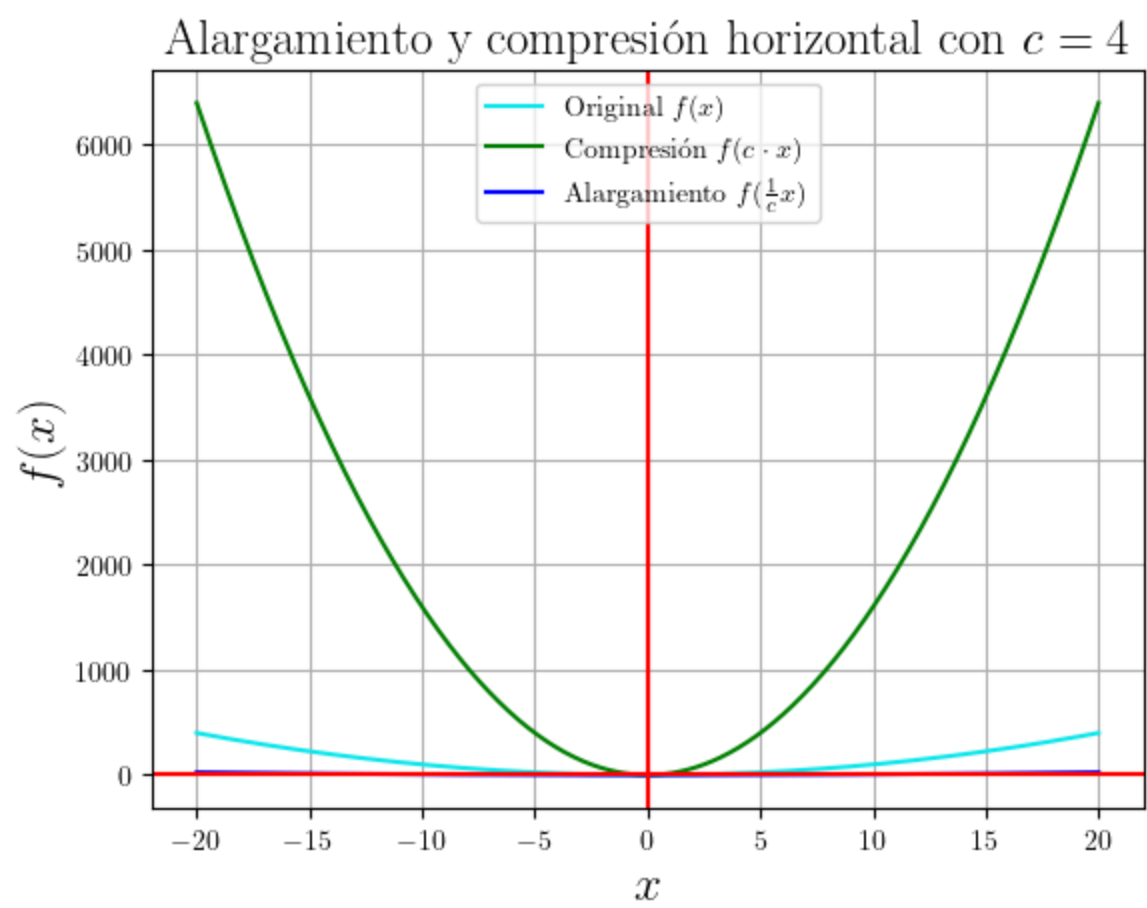
## Horizontal

- $y = f(c \cdot x)$  comprime la gráfica horizontalmente en un factor de  $c$ .
- $y = f(\frac{1}{c} \cdot x)$  alarga la gráfica horizontalmente en un factor de  $c$ .

```
In [ ]: # definiendo funcion parabola
def f(x):
    return x**2

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-20,20, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original $f(x)$',color='#00EE9ED')
ax.plot(x,f(c*x),label=r'Compresión $f(c\cdot x)$',color='green')
ax.plot(x,f((1/c)*x),label=r'Alargamiento $f(\frac{1}{c} x)$',color='blue')
ax.grid()
ax.axhline(y=0, color='r')
ax.axvline(x=0, color='r')
plt.title(r'Alargamiento y compresión horizontal con $c=4$',fontsize=18)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



## NOTA

Debido a que la parábola no es buena para representar este tipo de comportamiento, usaré una función  $seno(x)$  con el mismo código planteado, lo único que se le cambiará es el retorno del valor de `f(x)`

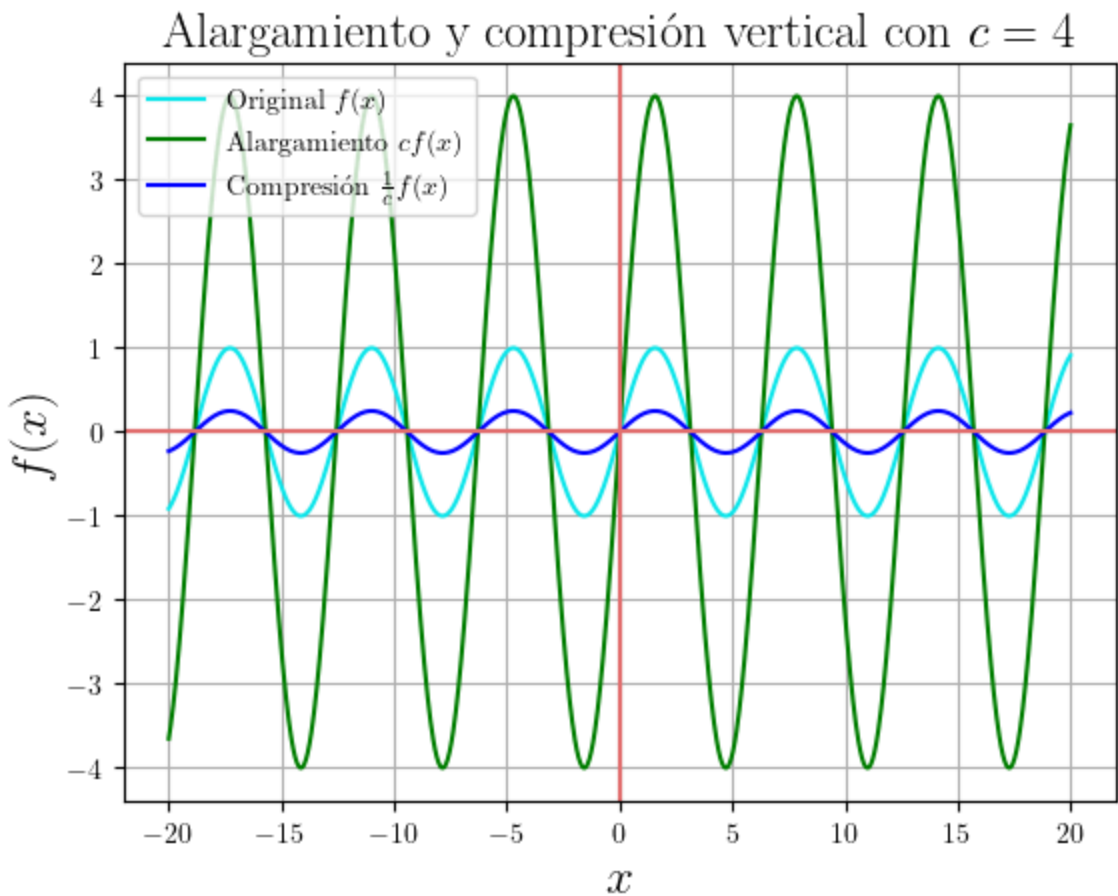
## Vertical

- $y = c \cdot f(x)$  alarga la gráfica verticalmente en un factor de  $c$ .
- $y = \frac{1}{c} \cdot f(x)$  comprime la gráfica verticalmente en un factor de  $c$ .

```
In [ ]: # definiendo funcion seno
def f(x):
    return np.sin(x)

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-20,20, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original  $f(x)$ ',color='#0EE9ED')
ax.plot(x,c*f(x),label=r'Alargamiento  $cf(x)$ ',color='green')
ax.plot(x,(1/c)*f(x),label=r'Compresión  $\frac{1}{c} f(x)$ ',color='blue')
ax.grid()
ax.axhline(y=0, color='#E16E6E')
ax.axvline(x=0, color='#E16E6E')
plt.title(r'Alargamiento y compresión vertical con  $c=4$ ',fontsize=18)
plt.xlabel(r' $x$ ',fontsize=18)
plt.ylabel(r' $f(x)$ ',fontsize=18)
plt.legend()
plt.show()
```



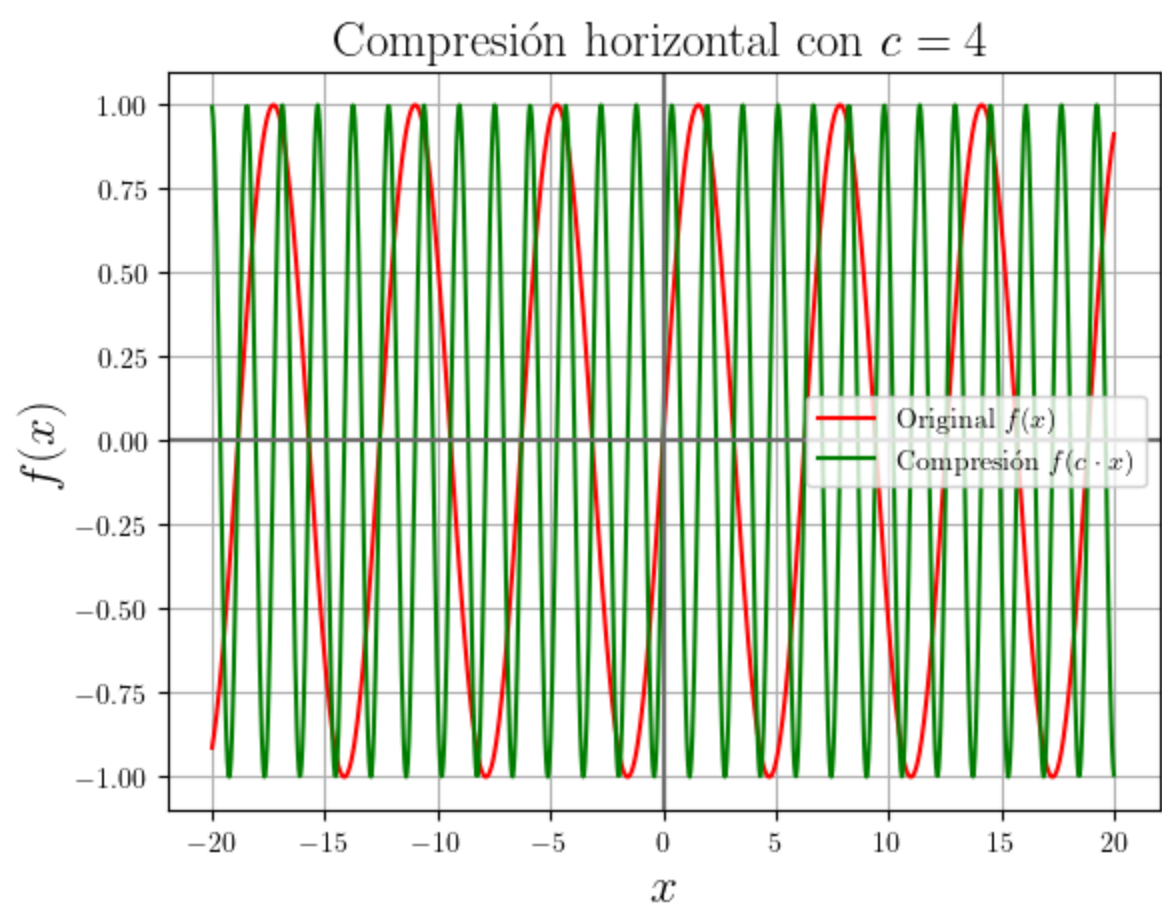
## Compresión horizontal

- $y = f(c \cdot x)$  comprime la gráfica horizontalmente en un factor de  $c$ .

```
In [ ]: # definiendo funcion seno(x)
def f(x):
    return np.sin(x)

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-20,20, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original  $f(x)$ ',color='red')
ax.plot(x,f(c*x),label=r'Compresión  $f(c \cdot x)$ ',color='green')
ax.grid()
ax.axhline(y=0, color='#727272')
ax.axvline(x=0, color='#727272')
plt.title(r'Compresión horizontal con  $c=4$ ',fontsize=18)
plt.xlabel(r' $x$ ',fontsize=18)
plt.ylabel(r' $f(x)$ ',fontsize=18)
plt.legend()
plt.show()
```



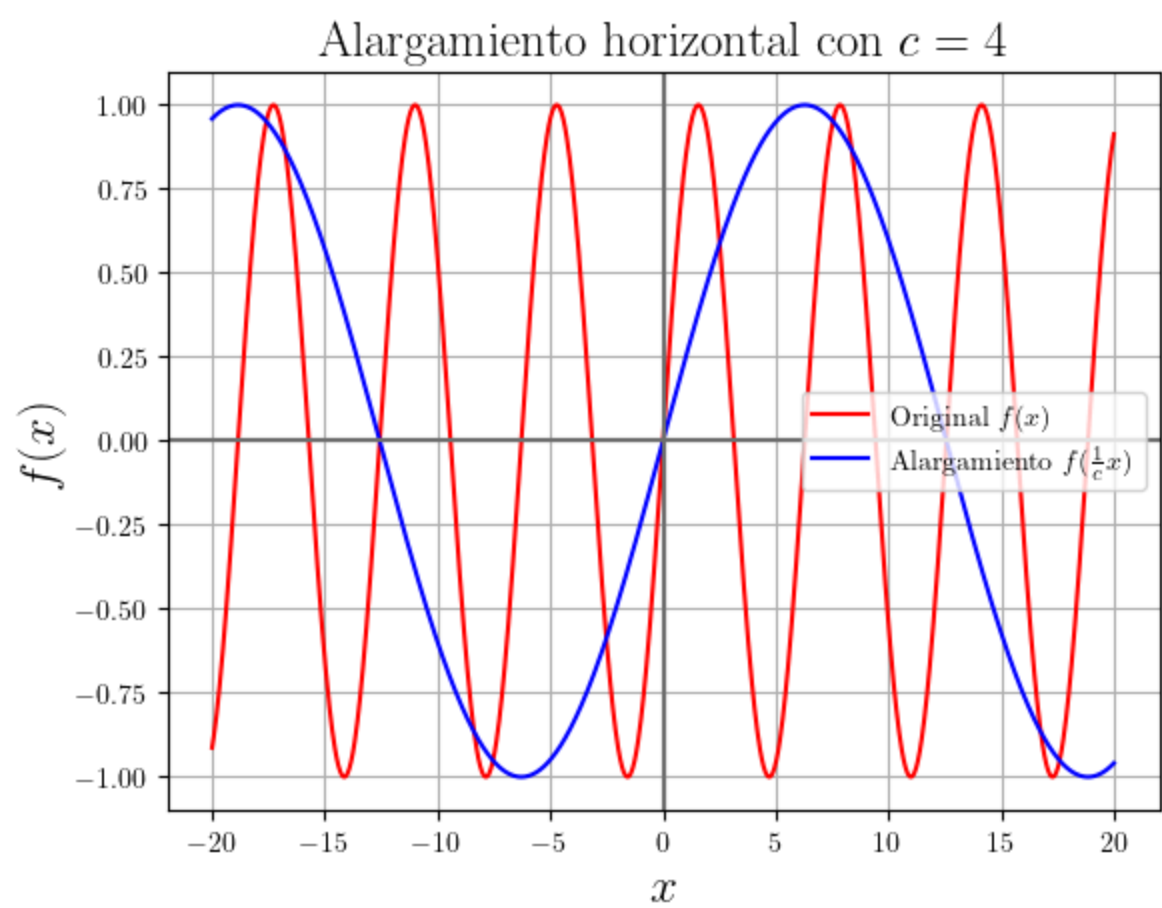
## Alargamiento horizontal

- $y = f(\frac{1}{c} \cdot x)$  alarga la gráfica horizontalmente en un factor de  $c$ .

```
In [ ]: # definiendo funcion seno(x)
def f(x):
    return np.sin(x)

#Definiendo variables
N = 1000
c = 4
x = np.linspace(-20,20, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original  $f(x)$ ',color='red')
ax.plot(x,f((1/c)*x),label=r'Alargamiento  $f(\frac{1}{c} x)$ ',color='blue')
ax.grid()
ax.axhline(y=0, color='#727272')
ax.axvline(x=0, color='#727272')
plt.title(r'Alargamiento horizontal con  $c=4$ ',fontsize=18)
plt.xlabel(r' $x$ ',fontsize=18)
plt.ylabel(r' $f(x)$ ',fontsize=18)
plt.legend()
plt.show()
```



## Reflexiones

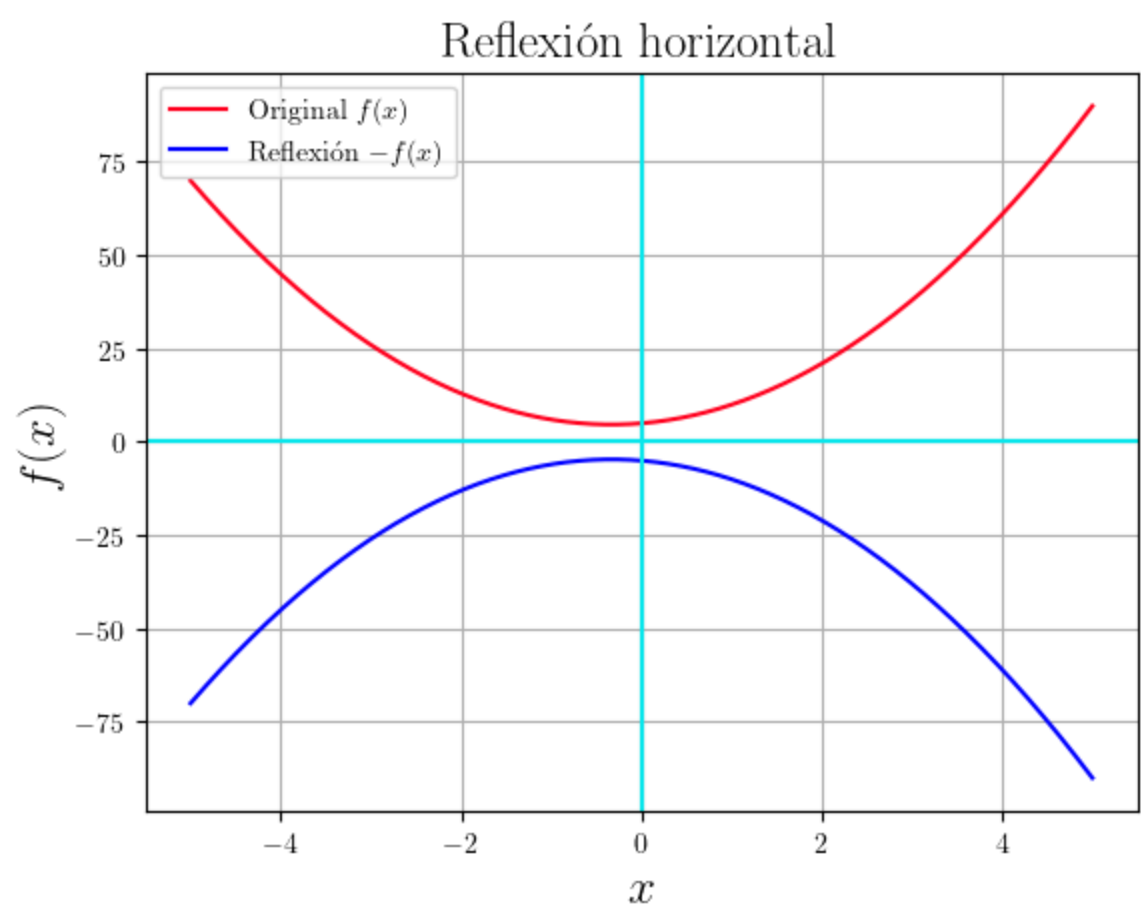
### Horizontal

- $y = -f(x)$  refleja la gráfica respecto al eje x.

```
In [ ]: # definiendo funcion lineal
def f(x):
    y=( 3*(x**2) + (2*x) + 5)
    return y

#Definiendo variables
N = 1000
x = np.linspace(-5,5, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original $f(x)$',color='#FF001E')
ax.plot(x,-1*f(x),label=r'Reflexión $-f(x)$',color='blue')
ax.grid()
ax.axhline(y=0, color='#0EE9ED')
ax.axvline(x=0, color='#0EE9ED')
plt.title(r'Reflexión horizontal',fontsize=18)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



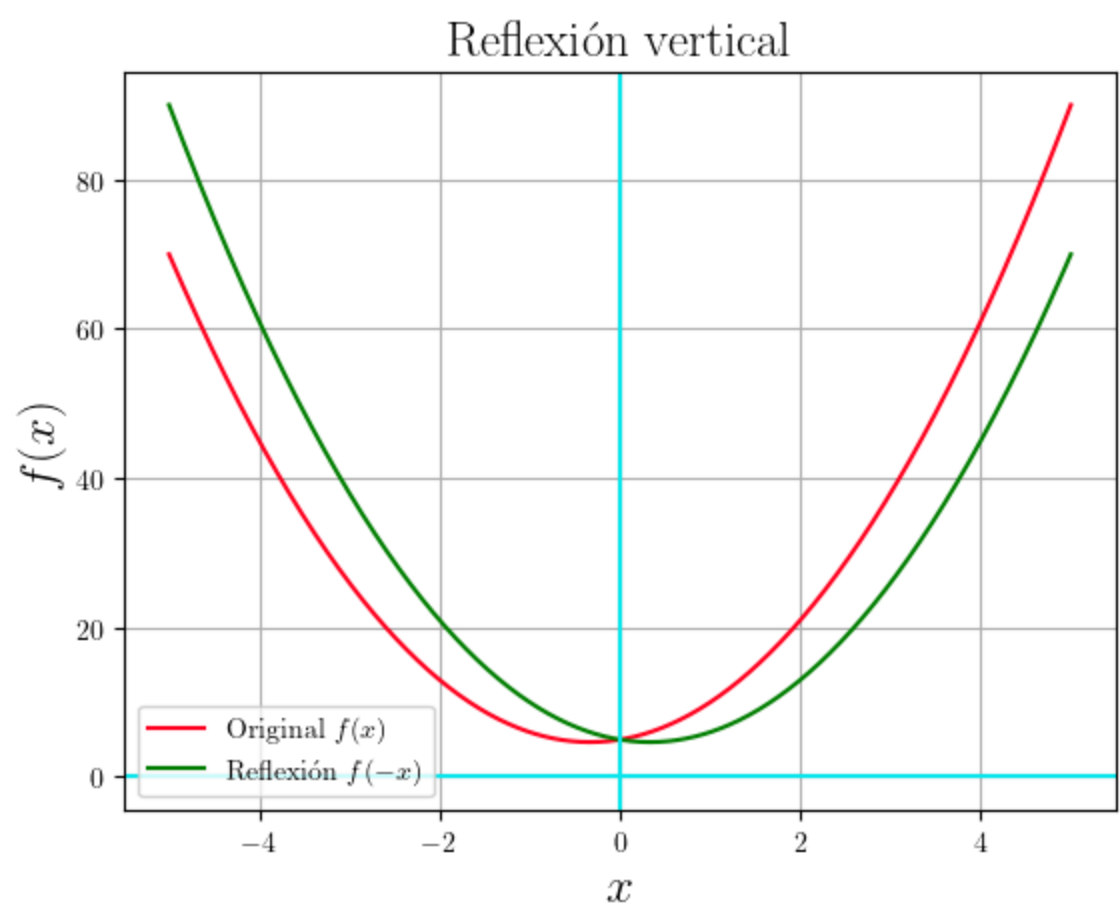
Vertical

- $y = f(-x)$  refleja la gráfica respecto al eje y.

```
In [ ]: # definiendo funcion lineal
def f(x):
    y=( 3*(x**2) + (2*x) + 5)
    return y

#Definiendo variables
N = 1000
x = np.linspace(-5,5, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original $f(x)$',color='#FF001E')
ax.plot(x,f(-1*x),label=r'Reflexión $f(-x)$',color='green')
ax.grid()
ax.axhline(y=0, color='#0EE9ED')
ax.axvline(x=0, color='#0EE9ED')
plt.title(r'Reflexión vertical ',fontsize=18)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```

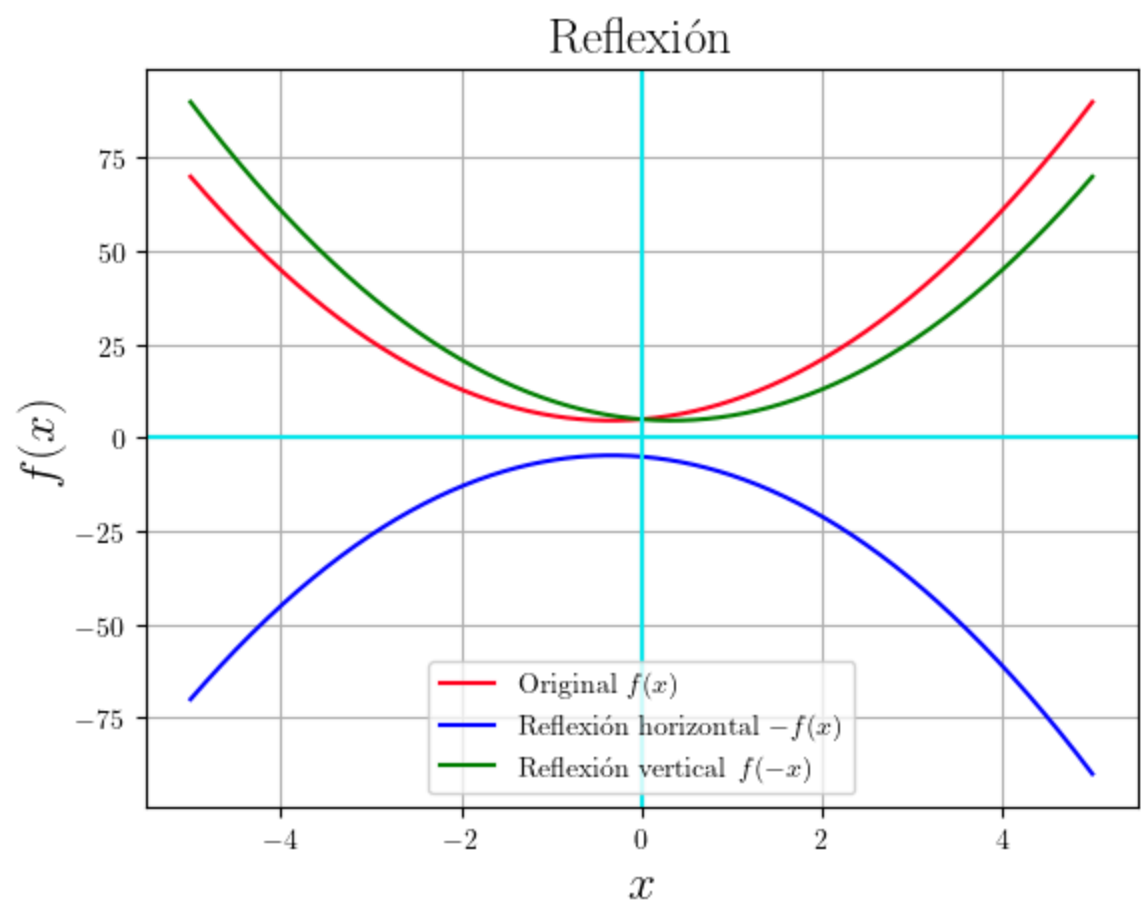


## Comparación de ambos

```
In [ ]: # definiendo funcion lineal
def f(x):
    y=( 3*(x**2) + (2*x) + 5)
    return y

#Definiendo variables
N = 1000
x = np.linspace(-5,5, num=N)

#Creando graficas de Compresion y alargamiento
fig, ax = plt.subplots()
ax.plot(x,f(x),label=r'Original $f(x)$',color='#FF001E')
ax.plot(x,-1*f(x),label=r'Reflexión horizontal $-f(x)$',color='blue')
ax.plot(x,f(-1*x),label=r'Reflexión vertical $f(-x)$',color='green')
ax.grid()
ax.axhline(y=0, color='#0EE9ED')
ax.axvline(x=0, color='#0EE9ED')
plt.title(r'Reflexión ',fontsize=18)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



## Explicación:

Cuando hablamos de reflexión, es bueno verlo como:

- Vertical: Gira sobre el **eje y**.
- Horizontal: Gira sobre el **eje x**.