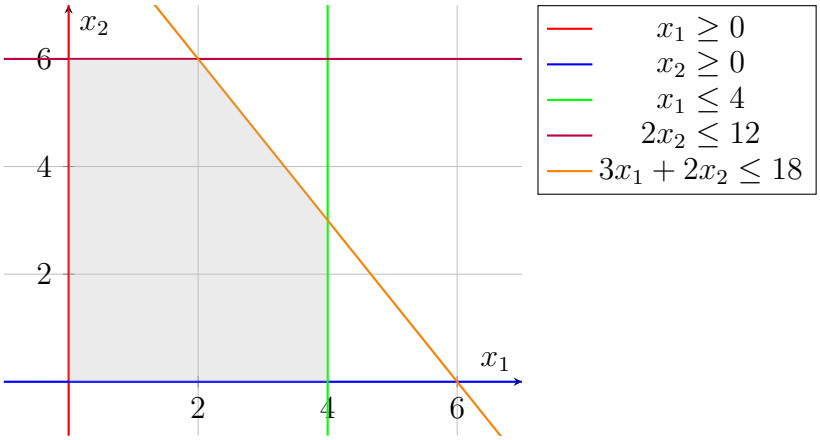
Programación No Lineal

Investigación Operativa, Universidad de San Andrés

Si encuentran algún error en el documento o hay alguna duda, mandenmé un mail a rodriguezf@udesa.edu.ar y lo revisamos.

1. Introducción a la Programación No Lineal

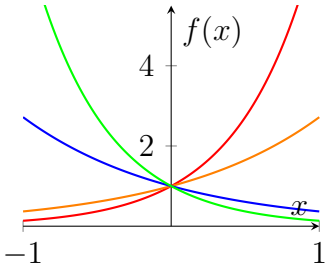
Hasta ahora veníamos viendo problemas solo de programación lineal, en donde tanto la función objetivo como las restricciones eran lineales (o dicho de forma más simple, todo son rectas). Por ejemplo, teníamos gráficos de este estilo:



Ahora vamos a tener al menos una de las dos cosas que no es lineal, que puede ser o bien la función objetivo o bien alguna de las restricciones.

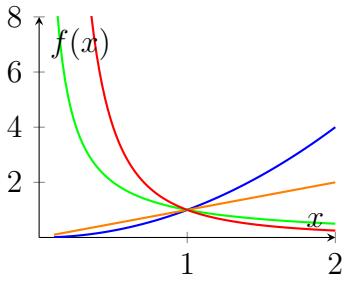
2. Convexidad

La convexidad es un concepto central en optimización. Una función es convexa si la recta que une dos puntos cualesquiera de su gráfica queda por encima o sobre la función. Esto tiene consecuencias muy importantes en problemas de optimización.



Funciones Exponenciales

$f(x) = e^{\alpha x} \quad \forall \alpha \in \mathbb{R}$



Funciones Potenciales

$f(x) = x^{\alpha} \qquad x \geq 0 \qquad \alpha \leq 0 \ \vee \ \alpha \geq 1$

Propiedades:

- La suma de dos funciones **convexas** es convexa.
- Un problema de optimización es convexo si y solo si:
  1. El conjunto factible es convexo
  2. El objetivo es minimizar una función convexa (o maximizar una cóncava)
- Si un problema de optimización es convexo, **cualquier óptimo local es un óptimo global**. De otra forma, alcanza con encontrar *algún* mínimo local que sabremos que es global.

3. Ejemplos

3.1. Asignación de presupuesto en publicidad digital

Una empresa desea asignar su presupuesto diario de publicidad entre dos plataformas: **Google Ads** (x_1) e **Instagram Ads** (x_2). El objetivo es maximizar el impacto total de la campaña, medido como una **función no lineal del gasto** en cada plataforma, debido a efectos de saturación. La empresa dispone de \$10.000 para su inversión, y para que te dejen meter publicidades, se deben invertir al menos \$2.000 en Google Ads y \$1.000 en Instagram Ads.

La función objetivo ya la sabe la empresa y está basada en datos históricos que relacionan el gasto con el impacto:

$$f(x_1, x_2) = -\left(100 \cdot \left(1 - e^{-0{,}05x_1}\right) + 80 \cdot \left(1 - e^{-0{,}08x_2}\right)\right)$$

Nota: La mayoría de los métodos de optimización numérica van a buscar siempre minimizar una función. Cuando querramos maximizar vamos a tener que agregarle un "-" al principio.

Las restricciones sí las tenemos que plantear nosotros, y en este caso son:

- $x_1 + x_2 \leq 10\,000$ (se dispone de \$10.000 para invertir)
- $x_1 \geq 2\,000$ (se debe invertir al menos \$2.000 en Google Ads)
- $x_2 \geq 1\,000$ (se debe invertir al menos \$1.000 en Instagram Ads)

¿Es convexo?

3.2. Ejemplo 2: Producción óptima de dos productos

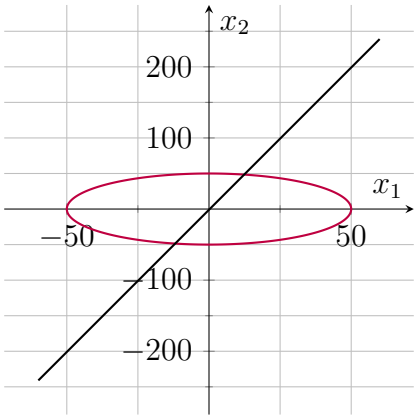
Una empresa fabrica dos productos, **A** y **B**, con ganancias unitarias de \$40 y \$30 respectivamente. La empresa desea determinar cuántas unidades de cada producto debe fabricar por día para **maximizar su ganancia diaria**, enfrentando restricciones no lineales:

- **Capacidad de máquinas:** $x_1^2 + x_2^2 \leq 2500$
- **Compatibilidad:** $\dfrac{x_1}{x_2 + 1} \leq 4$
- **No negatividad:** $x_1 \geq 0$, $x_2 \geq 0$

Función objetivo:

$$\text{máx}\, f(x_1, x_2) = 40x_1 + 30x_2$$

¿Es no lineal? ¿Convexa?



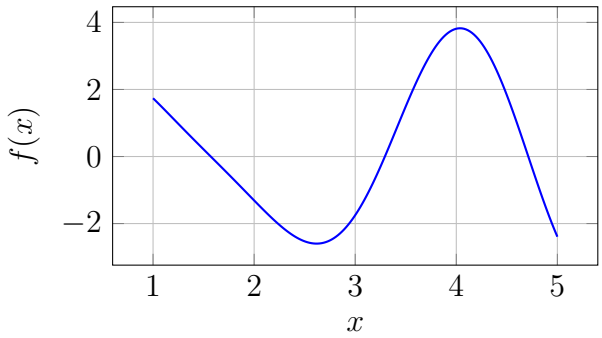
Resolución en Python:

```python
# Funcion objetivo (negativa para maximizar)
def ganancia_negativa(x):
    x1, x2 = x
    return -(40 * x1 + 30 * x2)

# Restriccion 1: x1^2 + x2^2 <= 2500
def restriccion_maquina(x):
    return 2500 - (x[0]**2 + x[1]**2)

# Restriccion 2: x1 / (x2 + 1) <= 4
def restriccion_compatibilidad(x):
    return 4 * (x[1] + 1) - x[0]

# Lista de restricciones
restricciones = [
    {'type': 'ineq', 'fun': restriccion_maquina},
    {'type': 'ineq', 'fun': restriccion_compatibilidad}
]

# Limites (x1, x2 >= 0)
bounds = [(0, None), (0, None)]

# Valor inicial factible
x0 = [1, 1]

# Optimizacion
from scipy.optimize import minimize
res = minimize(ganancia_negativa, x0, method='SLSQP', bounds=bounds,
    constraints=restricciones)

```

```python
# Mostrar resultado
if res.success:
    x1_opt, x2_opt = res.x
    ganancia_max = -res.fun
    print(f'Produccion optima de A: {x1_opt:.2f} unidades')
    print(f'Produccion optima de B: {x2_opt:.2f} unidades')
    print(f'Ganancia maxima: ${ganancia_max:.2f}')
else:
    print('Error:', res.message)
```

3.3. Ejemplo 3: Múltiples mínimos locales

En problemas no lineales, puede haber varios mínimos locales. El resultado depende del punto inicial:



Depende de donde empiece llego a un mínimo diferente. Para resolver esto usamos seeds (semillas aleatorias).

3.4. Ejemplo 4: Optimización en biotecnología

Una startup de biotecnología busca desarrollar empaques ecológicos usando dos ingredientes clave: **fibra vegetal** (x_1) y **alga marina** (x_2), ambos en kg por lote. El costo total depende de una función periódica y lineal de ambos ingredientes, más un costo fijo:

$$f(x_1, x_2) = \sin(x_1) \cdot \cos(x_2) + 0{,}1(x_1 + x_2) + 25$$

Restricciones:

- $x_1 + x_2 \geq 2$ (mínimo de ingredientes)
- $x_1 + 2x_2 \leq 8$ (capacidad máxima)
- $0 \leq x_1 \leq 6$
- $0 \leq x_2 \leq 6$

Resolución en Python:

```python
import numpy as np
from scipy.optimize import minimize

def costo(x):
    x1, x2 = x
    return np.sin(x1) * np.cos(x2) + 0.1 * (x1 + x2) + 25

restricciones = [
    {'type': 'ineq', 'fun': lambda x: x[0] + x[1] - 2},
    {'type': 'ineq', 'fun': lambda x: 8 - (x[0] + 2 * x[1])}
]

bounds = [(0, 6), (0, 6)]

np.random.seed(42)
resultados = []
for i in range(10):
    x0 = np.random.uniform(0, 6, size=2)
    res = minimize(costo, x0, method='SLSQP', bounds=bounds,
    constraints=restricciones)
    if res.success:
        resultados.append((res.fun, res.x))

resultados.sort()
mejor_valor, mejor_x = resultados[0]
print(f'Mejor solucion encontrada: x1 = {mejor_x[0]:.4f}, x2 = {
    mejor_x[1]:.4f}')
print(f'Costo minimo estimado: {mejor_valor:.4f}')
```

4. Referencias

Lista de referencias o sugerencias para profundizar.