**FCEIA - Universidad Nacional de Rosario**
**Optimización Continua - LCC 2025**
**Profesor: Alejandro G. Marchetti**

---

# Guía de Ejercicios No. 4

**1)** Encuentre mínimos locales de la función $f(x) = (x_2^2 - x_1)^2$.

**2)** Encuentre puntos que satisfagan condiciones necesarias para máximos y mínimos locales de la función

$$f(x) = \frac{x_1 + x_2}{3 + x_1^2 + x_2^2 + x_1 x_2}$$

Intente establecer la naturaleza de dichos puntos verificando condiciones suficientes de optimalidad. (Recomendado: utilizar el Toolbox simbólico de Matlab o SymPy para facilitar los cálculos - Ver anexo).

**3)** Considere la función $f(x) = xe^{-2x}$. Encuentre todos los puntos estacionarios. ¿Puede asegurar algo acerca de un mínimo global o máximo global de $f(x)$? Justificar analíticamente.

**4)** Considere el problema $\min \|\mathsf{A}\mathbf{x} - \mathbf{b}\|^2$, donde $\mathsf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, y $\|\cdot\|$ es la norma euclídea en $\mathbb{R}^m$.

a) Dar la interpretación geométrica del problema.

b) Escribir la condición de optimalidad necesaria. ¿Es también suficiente?

c) ¿La solución óptima es única? ¿Por qué?

d) ¿Puedes dar una forma cerrada de la solución óptima? Especificar cualquier suposición de debas hacer.

e) Resolver el problema con:

$$\mathsf{A} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

**Anexo: Matemática Simbólica en Matlab**

```
%Definimos las variables simbólicas x1 y x2 como reales:
x1 = sym('x1','real'); x2 = sym('x2','real')
x = [x1 x2]';

%Definimos la función:
f = (x1 + x2)/(3 + x1^2 + x2^2 + x1*x2)

df = jacobian(f,x)      %Obtenemos el gradiente de f
ddf = jacobian(df,x)    %Obtenemos la matriz hessiana de f

u1 = -5:0.05:5;
u2 = -5:0.05:5;

for i=1:length(u1)
    for j=1:length(u2)
        fun(i,j) = (u1(i) + u2(j))/(3 + u1(i)^2 + u2(j)^2 + u1(i)*u2(j));
    end
end

figure(1)
mesh(u1,u2,fun')

figure(2)
[C,h] = contour(u1,u2,fun',[-0.33 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.33]);
set(h,'ShowText','on')

%Evaluamos la matriz hessiana de f en el punto a = [1 2]'
x1 = 1;    x2 = 2;
Ha = eval(ddf)

%Verificamos los valores propios de Ha
eig(Ha)
```

**Anexo: Matemática Simbólica en Python**

```
import sympy as sp

# Definimos las variables simbólicas
x1, x2 = sp.symbols('x1 x2')
x = sp.Matrix([x1, x2])

# Definimos la función
f = (x1 + x2)/(3 + x1**2 + x2**2 + x1*x2)
```

```
# Obtenemos el gradiente
df = sp.Matrix([sp.diff(f, var) for var in x])

# Obtenemos la matriz hessiana
ddf = df.jacobian(x)

# Mostrar resultados simbólicos
print("Gradient:")
sp.pprint(df)
print("Hessian:")
sp.pprint(ddf)

# Evaluamos en el punto (x1, x2) = (1, 2)
a = x1: 1, x2: 2
gradval = df.subs(a)
hessval = ddf.subs(a)

print("Gradient at (1, 2):")
sp.pprint(gradval)
print("Hessian at (1, 2):")
sp.pprint(hessval)
```