COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

comillas.edu

comillas.edu

Nonlinear Optimization

Andrés Ramos (Andres.Ramos@comillas.edu)
Pedro Sánchez (Pedro.Sanchez@comillas.edu)
Sonja Wogrin (Sonja.Wogrin@comillas.edu)

Departamento de Organización Industrial

Contents

NLP PROBLEMS
TYPE OF NLP PROBLEMS
CLASSIFICATION OF UNCONSTRAINED METHODS
OPTIMALITY CONDONDITIONS FOR NONLINEAR UNCONSTRAINED OPTIMIZATION
OPTIMALITY CONDITIONS FOR NLP
METHODS FOR UNCONSTRAINED OPTIMIZATION (master)
NONLINEAR PROGRAMMING METHODS (master)

1

NLP PROBLEMS

Nonlinear programming problems (i)

- The transportation problem with volume discounts
  - The unitary price of transportation between two points decreases depending on the volume that is transported.
- The optimal power flow problem of an electric power system
  - Losses are nonlinear
- The product-mix problem with price and/or cost elasticity
  - The demand curve or price-demand curve $p(x)$ represents the unitary price that is needed to sell x units. This is a decreasing function, which is never below the unitary production cost c. The gross revenues (price times quantity produced) is a nonlinear expression. Gross profit margin

$$f(x) = xp(x) - cx$$

  - The nonlinear costs can appear due to an increased unitary efficiency depending on the quantity.

Nonlinear programming problems (ii)

- Selection of an investment portfolio

  n types of shares

  $x_j, j = 1, \dots, n$ represent the number of shares j which will be included in the portfolio

  μ_j and σ_{jj} are the historic mean and the variance of the output of each share type j, where σ_{jj} is a risk measure of these shares. Let σ_{ij} be the covariance of the output of a share type i and j.

  $R(x)$ is the expected output and its variance $V(x)$

$$R(x) = \sum_{j=1}^{n} \mu_j x_j$$

$$V(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

The objective function is $f(x) = R(x) - \beta V(x)$

where β is the risk aversion factor.

2

TYPE OF NLP PROBLEMS

Nonlinear optimization (i)

- Optimization WITHOUT constraints

$$\min_x f(x)$$
$$f(x): \mathbb{R}^n \to \mathbb{R}$$
$$x \in \mathbb{R}^n$$

- Optimization WITH constraints (Nonlinear programming NLP)

$$\min_x f(x)$$
$$g_i(x) = 0 \quad i \in \varepsilon$$
$$g_i(x) \leq 0 \quad i \in \varphi$$
$$f(x): \mathbb{R}^n \to \mathbb{R}$$
$$g_i(x): \mathbb{R}^n \to \mathbb{R}$$
$$x \in \mathbb{R}^n$$

Nonlinear optimization (ii)

- Quadratic Programming

$$\min_x f(x) = \frac{1}{2} x^T Q x - b^T x$$
$$Ax = b$$

- Convex Programming

  $f(x)$ is convex (concave if we are maximizing) and $g_i(x)$ is convex, $\forall i = 1, \dots, m$

- Separable Programming

  The function can be separated into a sum of functions of the individual variables

$$f(x) = \sum_{j=1}^{n} f_j(x_j)$$

- Geometric Programming

  Objective function and constraints take the form

$$g(x) = \sum_{j=1}^{n} c_j P_j(x)$$

$$P_j(x) = x_1^{a_{j1}} x_2^{a_{j2}} \dots x_n^{a_{jn}}, \qquad j = 1, \dots, n$$

3

CLASSIFICATION OF UNCONSTRINED METHODS

Classification of optimization methods WITHOUT constraints according to the use of derivatives

- Without derivatives
  - Necessary when derivatives cannot be calculated
- First derivatives (gradient)
- Second derivatives (Hessian)
  - Higher computational cost
  - Better convergence properties

Taylor series expansion

- Approximate a function f close to a given point x_0
- It is necessary to know the derivatives of the function

$$f(x_0 + p) = f(x_0) + \nabla f(x_0)^T p + \frac{1}{2} p^T \nabla^2 f(x_0) p + \cdots$$

$p \in \mathbb{R}^n$ is a vector different from 0

$f(x)$ is the value of the function

$\nabla f(x)$ is the gradient of the function

$\nabla^2 f(x)$ is the Hessian of the function (if f has continuous second derivatives, then this is a symmetric matrix)

- Or alternatively

$$f(x_0 + p) = f(x_0) + \nabla f(x_0)^T p + \frac{1}{2} p^T \nabla^2 f(\xi) p$$

where ξ is a point between x and x_0

Quadratic function (i)

$$f(x) = \frac{1}{2}x^T Q x - b^T x$$
$$x \in \mathbb{R}^n$$
$$b \in \mathbb{R}^n$$
$$Q \in \mathbb{R}^n \times \mathbb{R}^n$$

- Gradient

$$\nabla f(x) = Qx - b$$

- Hessian

$$\nabla^2 f(x) = Q$$

Quadratic function (ii)

$$f(x,y) = \frac{1}{2}x^2 + 2xy + \frac{1}{2}y^2 - y + 9$$

$$\nabla f(x,y) = \begin{pmatrix} x + 2y \\ 2x + y - 1 \end{pmatrix}$$

$$\nabla^2 f(x,y) = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

- We are in $x_0 = (1, -1)$ $f(1, -1) = 9$ and want to know the function value in $x_1 = (1.1, -0.9)$
- Direct evaluation

$$f(1.1, -0.9) = 0.605 - 1.98 + 0.405 + 0.9 + 9 = 8.93$$

- Approximation using the Taylor series expansion

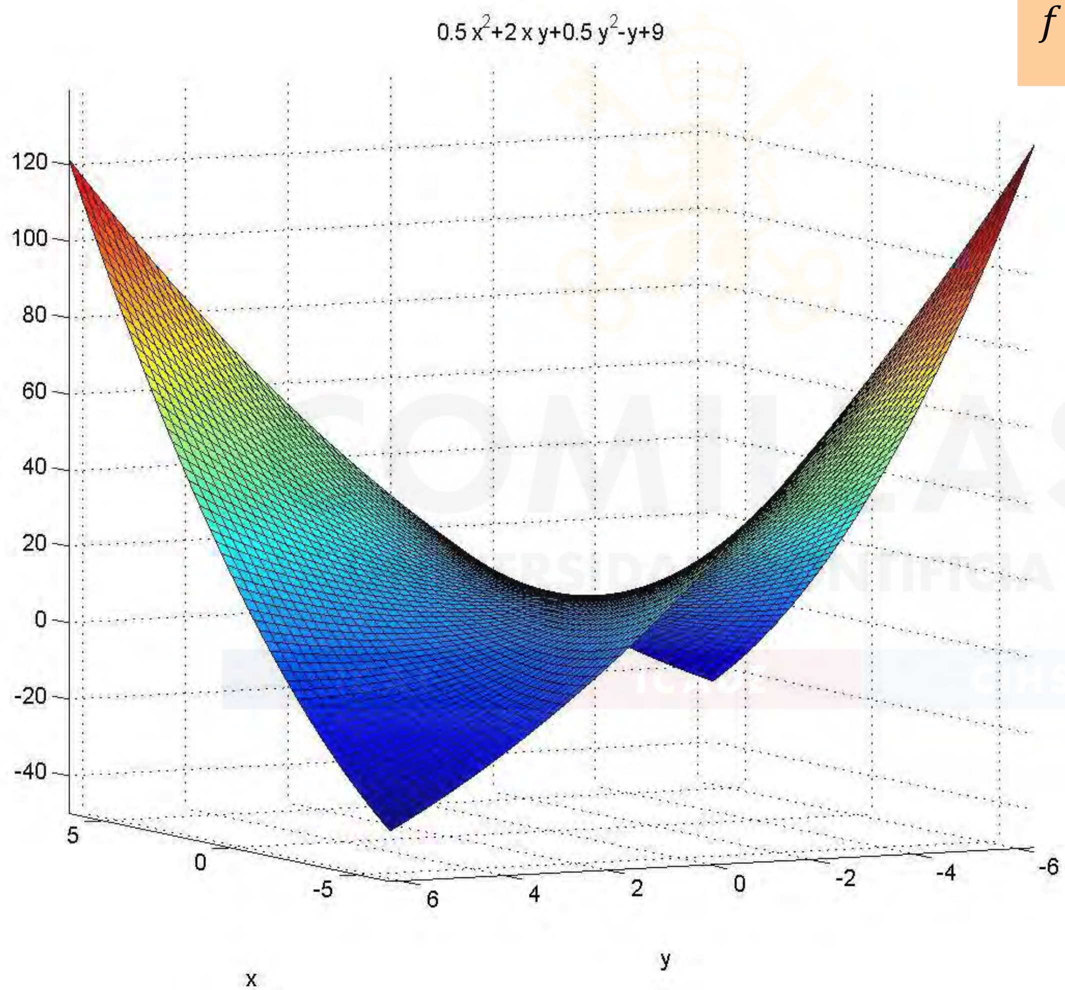
$$f(1.1, -0.9) = 9 + (-1 \quad 0)\begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} + \frac{1}{2}(0.1 \quad 0.1)\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} = 9 - 0.1 + 0.03 = 8.93$$

- For a quadratic function, the 2nd order Taylor series expansion is exact

Quadratic function (iii)

```
ezsurf('0.5*x^2+2*x*y+0.5*y^2-y+9')
```

$$f(x,y) = \frac{1}{2}x^2 + 2xy + \frac{1}{2}y^2 - y + 9$$



Local, global minimum (i)

- Let the function f be continuously differentiable for the first and second order

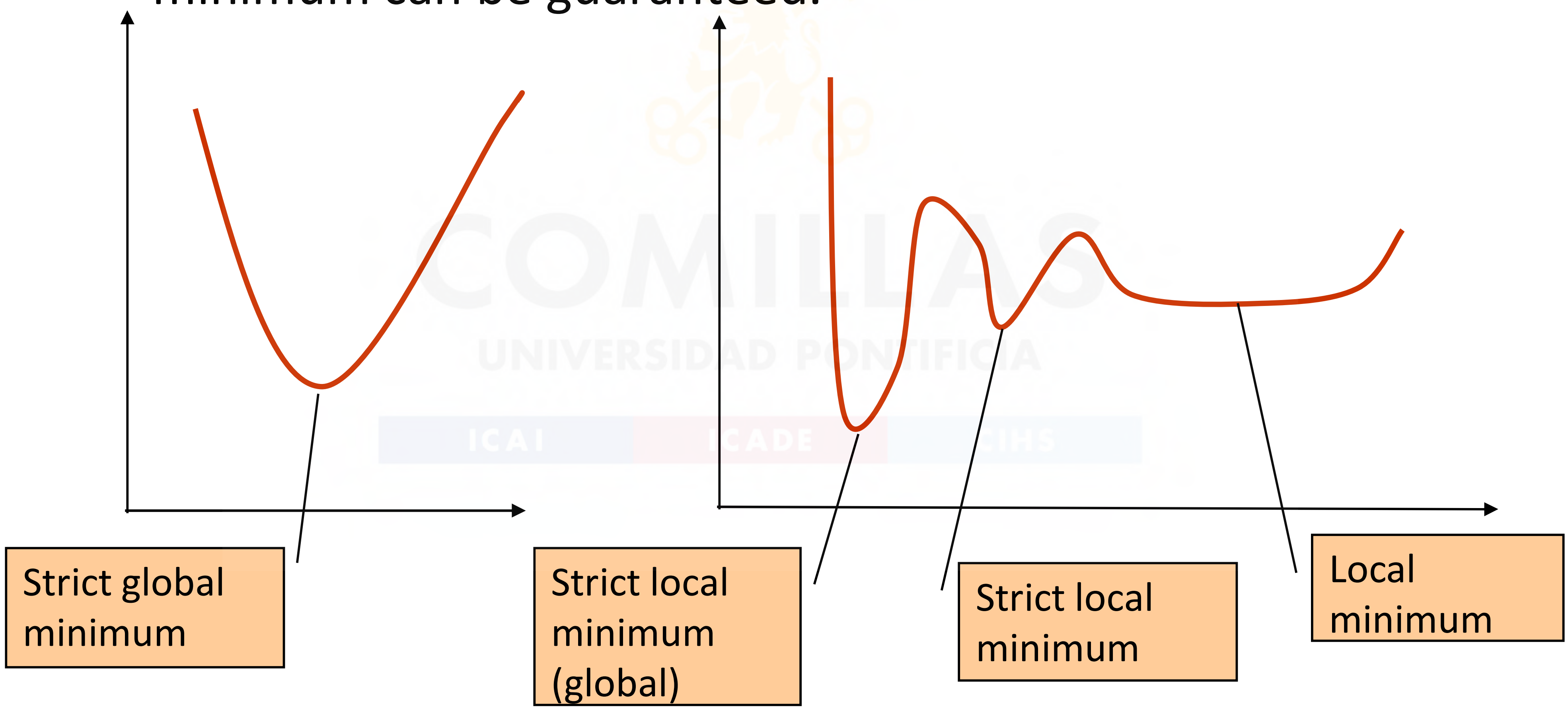
$$\min_x f(x)$$
$$f(x): \mathbb{R}^n \to \mathbb{R}$$
$$x \in \mathbb{R}^n$$

$x^* \in \mathbb{R}^n$ is the optimum of the function

- It is a global minimum if $f(x^*) \le f(x)$ for $x \in \mathbb{R}^n$
- It is a strict global minimum if $f(x^*) < f(x)$ for $x \in \mathbb{R}^n$
- It is a local minimum if $f(x^*) \le f(x)$ in its vicinity $\|x - x^*\| < \varepsilon$ where ε is a positive number (typically small) whose exact value can depend on x^*
- It is a strict local minimum if $f(x^*) < f(x)$ in its vicinity $\|x - x^*\| < \varepsilon$

Local and global minima (ii)

- A global minimum is hard to find. Many methods are local.
- Only under additional assumptions (convexity) global minimum can be guaranteed.



4

OPTIMALITY CONDITIONS FOR NONLINEAR UNCONSTRAINED OPTIMIZATION

UNCONSTRAINED optimization
Optimality conditions (i)

$$\min_x f(x)$$
$$x \in \mathbb{R}^n$$

- First order necessary condition
  - If x^* is a local minimum of f then necessarily $\nabla f(x^*) = 0$
  - That condition is satisfied for any stationary point
- Second order necessary condition
  - If x^* is a local minimum of f then necessarily $\nabla^2 f(x^*)$ is a positive semidefinite matrix (equivalent to convex, positive curvature)
- Second order sufficient condition
  - If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite then x^* is a strict local minimum of f
- Necessary and sufficient condition
  - Let be f convex and differentiable in x^* (if it is twice differentiable Hessian is positive semidefinite); x^* is a local minimum if and only if $\nabla f(x^*) = 0$. x^* is a global minimum if and only if it is convex in all the region

Type of NLP solutions (iii)
Positive definite matrix

- A matrix A is positive definite if $x^T Ax > 0$ for all non-zero vectors x
- Or rather if all its eigenvalues are positive
- Or rather if all its leading principal minors (determinants of order $1,2,\ldots,n$ (being n the matrix dimension) obtained by adding consecutive rows and columns starting from the first element) are positive.

This matrix is positive definite $\nabla^2 f(x,y) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

This one is not $\nabla^2 f(x_1,x_2) = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$

Type of NLP solutions (iv)
Positive semidefinite (psd) matrix

- A matrix A is positive semidefinite if $x^T A x \geq 0$ for all non-zero vectors x
- Or rather if all its eigenvalues are nonnegative
- Or rather if all its first minors of order 1,2,…,n (being n the matrix dimension) are nonnegative
- First minors are the determinants of k arbitrary rows and their corresponding columns

$\delta_1 = |1| = 1$ $\quad \delta_2 = |12| = 12$ $\quad \delta_3 = |4| = 4$

$$\delta_{1,2} = \begin{vmatrix} 1 & -3 \\ -3 & 12 \end{vmatrix} = 3 \quad \delta_{1,3} = \begin{vmatrix} 1 & -1 \\ -1 & 4 \end{vmatrix} = 3 \quad \delta_{2,3} = \begin{vmatrix} 12 & 6 \\ 6 & 4 \end{vmatrix} = 12$$

$$\delta_{1,2,3} = \begin{vmatrix} 1 & -3 & -1 \\ -3 & 12 & 6 \\ -1 & 6 & 4 \end{vmatrix} = 0$$

❑ This matrix is positive semidefinite

$$\nabla^2 f(x,y,z) = \begin{pmatrix} 1 & -3 & -1 \\ -3 & 12 & 6 \\ -1 & 6 & 4 \end{pmatrix}$$

Type of NLP solutions (v)
Negative definite matrix

- A matrix A is negative definite if $x^T Ax < 0$ for all non-zero vectors x
- Or rather if all its eigenvalues are negative
- Or rather if all its leading minors are, alternatively, negative, positive, etc.
- Or rather is the matrix $-A$ (which is obtained by changing the sign of all the matrix elements) is positive definite.

Equivalent relations

Convex Function	Concave Function
Positive definite Hessian matrix	Negative definite Hessian matrix
Positive curvature (2^{nd} derivative)	Negative curvature (2^{nd} derivative)

Local minimum + convexity of the entire region	Local maximum + concavity of the entire region
Global Minimum	Global Maximum

Positive semidefinite (negative)	Positive definite (negative)
Local minimum (maximum)	Strict local minimum (maximum)

Example 1: Optimization WITHOUT constraints

$$f(x,y) = \frac{1}{2}x^2 + 2xy + \frac{1}{2}y^2 - y + 9$$

- Gradient = 0 yields a system of equations

$$\nabla f(x,y) = \begin{pmatrix} x + 2y \\ 2x + y - 1 \end{pmatrix} = 0$$

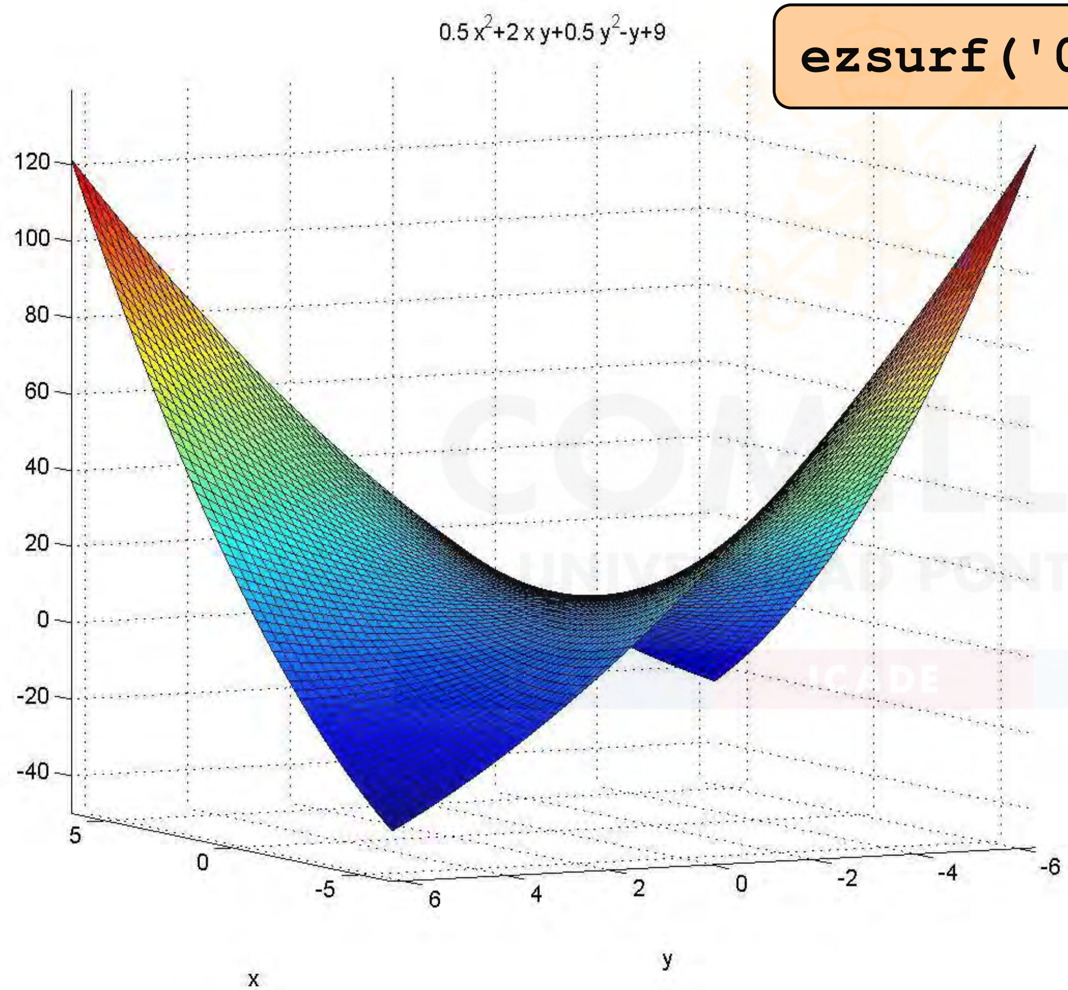
with solution $(x,y) = (\frac{2}{3}, -\frac{1}{3})$

- Hessian $\nabla^2 f(x,y) = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ is an indefinite matrix (it is not positive semidefinite nor negative semidefinite)

This means that it is neither a local maximum nor minimum

Example 1: Optimization WITHOUT constraints

$$f(x,y) = \frac{1}{2}x^2 + 2xy + \frac{1}{2}y^2 - y + 9$$



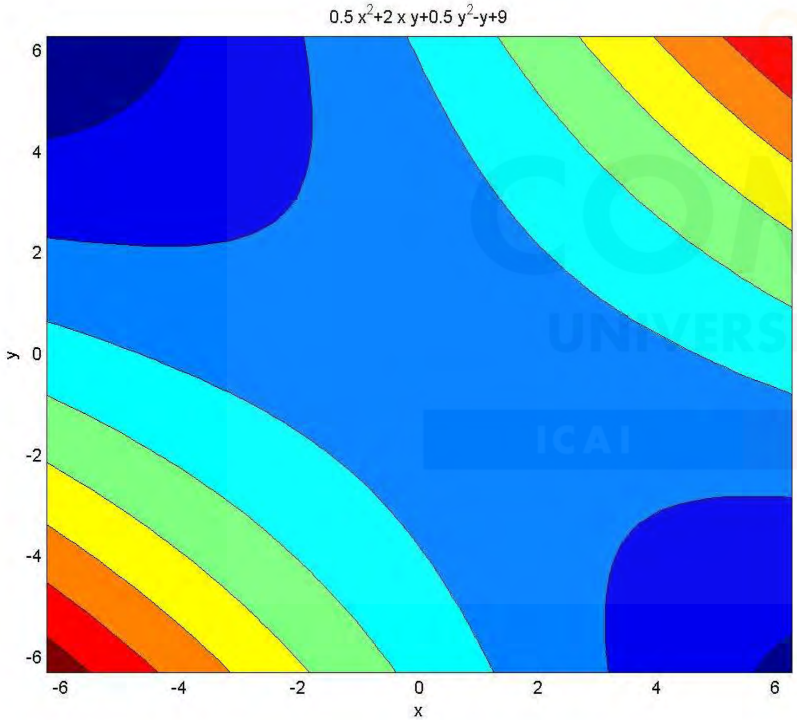

```
ezsurf('0.5*x^2+2*x*y+0.5*y^2-y+9')
```

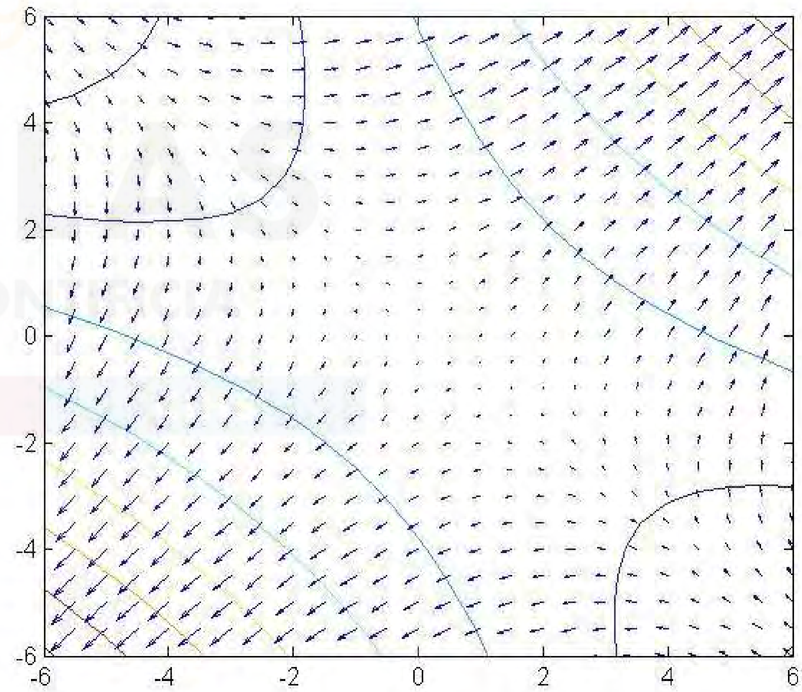
Example 1: Optimization WITHOUT constraints

$$f(x,y) = \frac{1}{2}x^2 + 2xy + \frac{1}{2}y^2 - y + 9$$

```
ezcontourf('0.5*x^2+2*x*y+0.5*y^2-y+9')
```

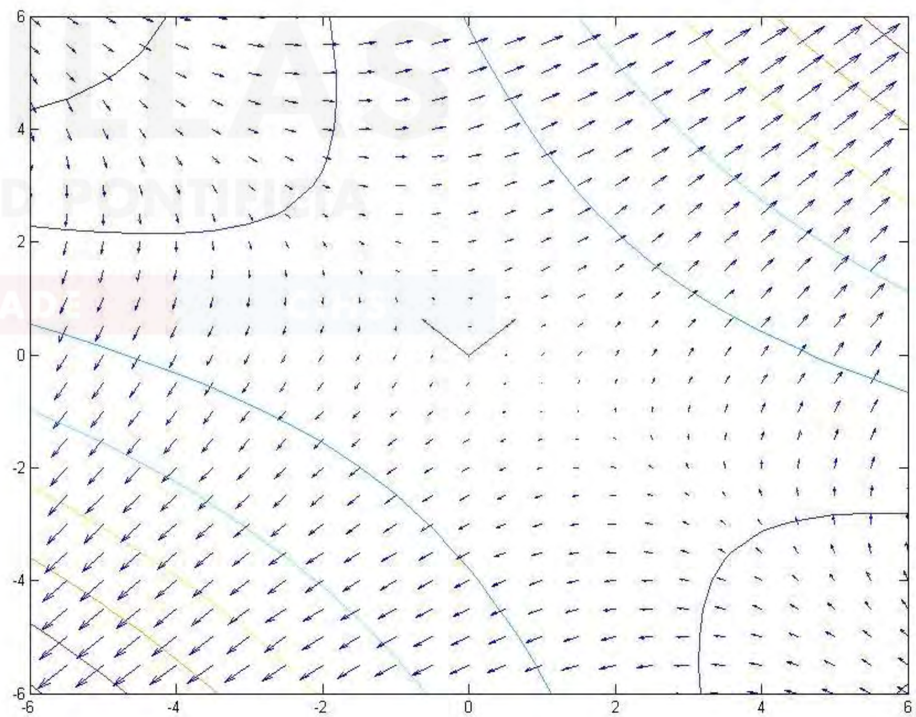
```
[x,y] = meshgrid(-6:0.5:6,-6:0.5:6);
z=0.5*x.^2+2*x.*y+0.5*y.^2-y+9;
[px,py] = gradient(z,0.5,0.5);
contour(x,y,z); hold on
quiver(x,y,px,py)
```





Eigenvalues and eigenvectors

- Hessian $\nabla^2 f(x,y) = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ is an indefinite matrix
- Its eigenvalues are $\lambda = \begin{pmatrix} -1 \\ 3 \end{pmatrix}$
- Its eigenvectors are $v_1 = \begin{pmatrix} -0.71 \\ 0.71 \end{pmatrix}$ $v_2 = \begin{pmatrix} 0.71 \\ 0.71 \end{pmatrix}$



Example 2: Optimization WITHOUT constraints

$$f(x,y) = 2x^2 + y^2 - 2xy + 2x^3 + x^4$$

- Gradient = 0 yields a system of equations

$$\nabla f(x,y) = \begin{pmatrix} 4x - 2y + 6x^2 + 4x^3 \\ 2y - 2x \end{pmatrix} = 0$$

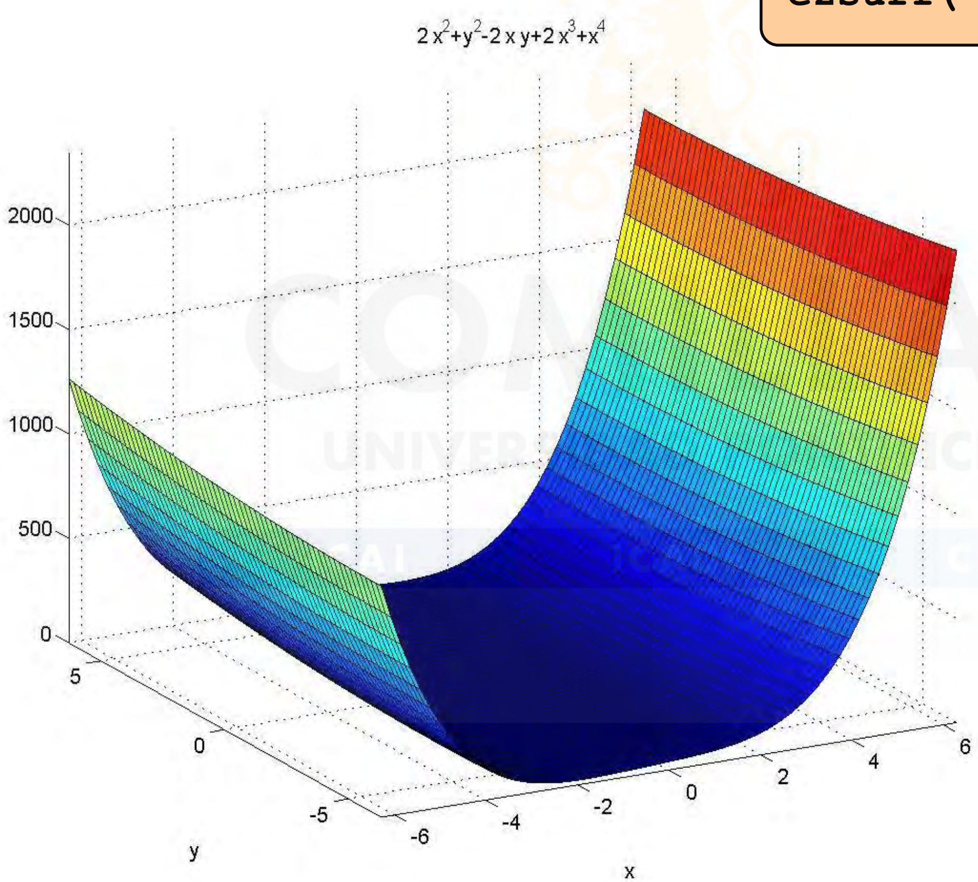
- Hessian

$$\nabla^2 f(x,y) = \begin{pmatrix} 4 + 12x + 12x^2 & -2 \\ -2 & 2 \end{pmatrix}$$

Example 2: Optimization WITHOUT constraints

$$f(x, y) = 2x^2 + y^2 - 2xy + 2x^3 + x^4$$

```
ezsurf('2*x^2+y^2-2*x*y+2*x^3+x^4')
```

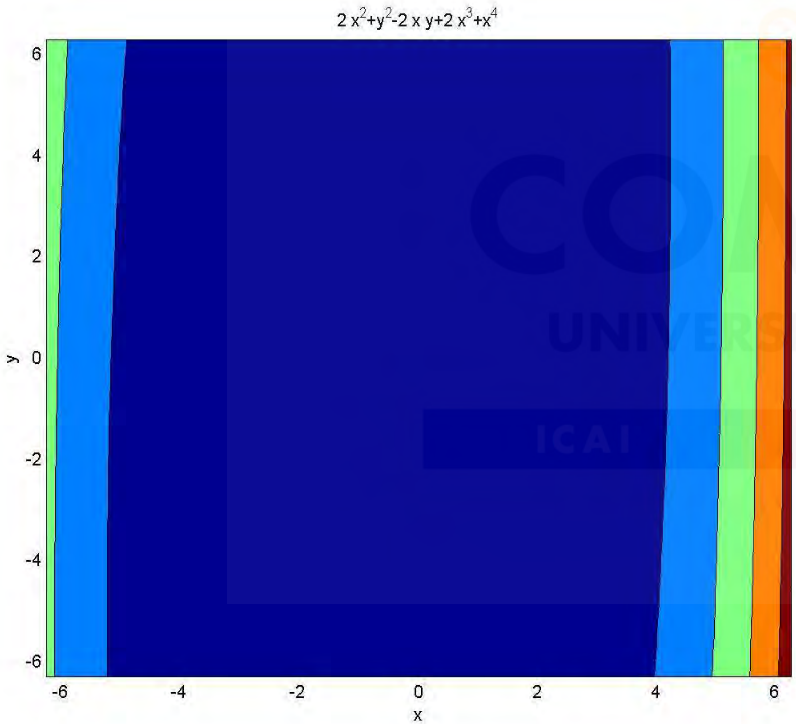


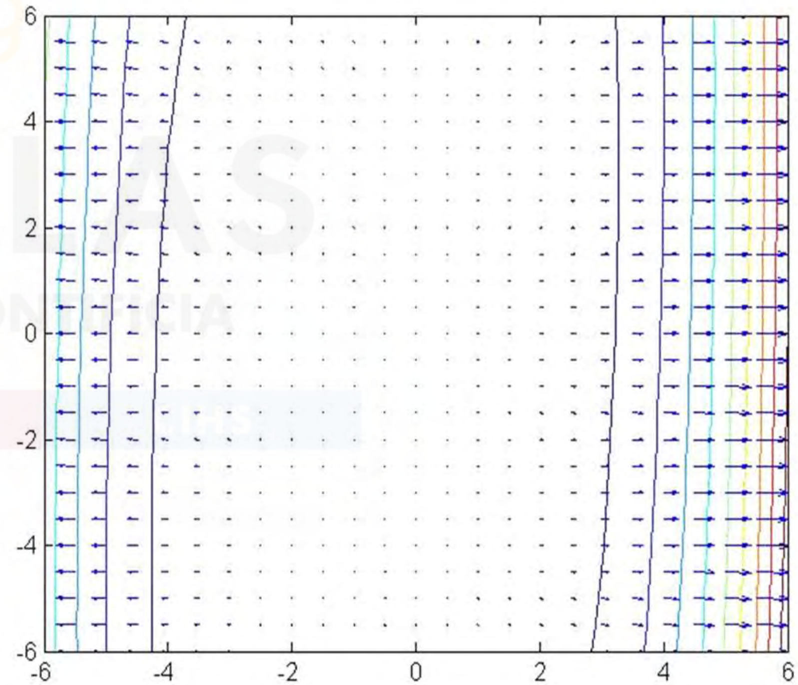
Example 2: Optimization WITHOUT constraints

$$f(x,y) = 2x^2 + y^2 - 2xy + 2x^3 + x^4$$

```
ezcontourf('2*x^2+y^2-2*x*y+2*x^3+x^4')
```

```
[x,y] = meshgrid(-6:0.5:6, -6:0.5:6);
z=2*x.^2+y.^2-2*x.*y+2*x.^3+x.^4;
[px,py] = gradient(z,0.5,0.5);
contour(x,y,z);
hold on, quiver(x,y,px,py)
```





Example 3: Optimization WITHOUT constraints

$$f(x,y) = (x-2)^2 + (y-1)^2$$

- Gradient = 0 yields a system of equations

$$\nabla f(x,y) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix} = 0$$

with solution $(x,y) = (2,1)$

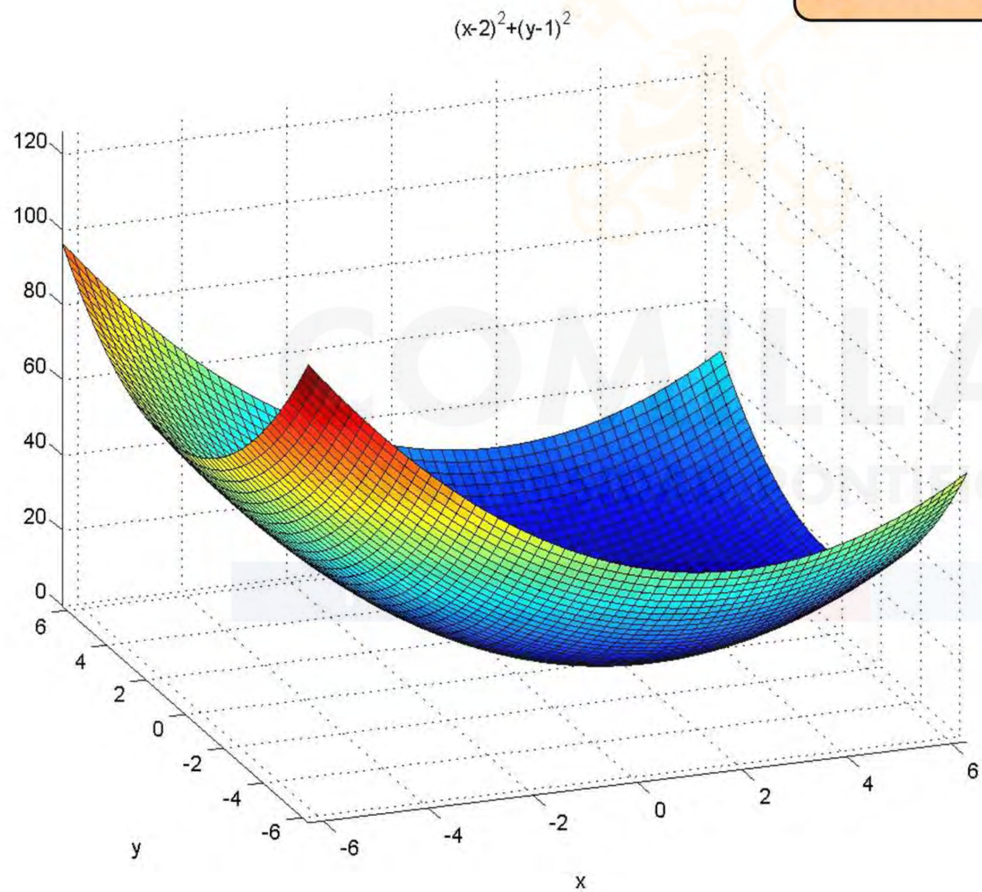
- Hessian $\nabla^2 f(x,y) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ is a positive definite matrix

and (independent of the point) this yields a global minimum

Example 3: Optimization WITHOUT constraints

$$f(x, y) = (x-2)^2 + (y-1)^2$$

```
ezsurf('(x-2)^2+(y-1)^2')
```

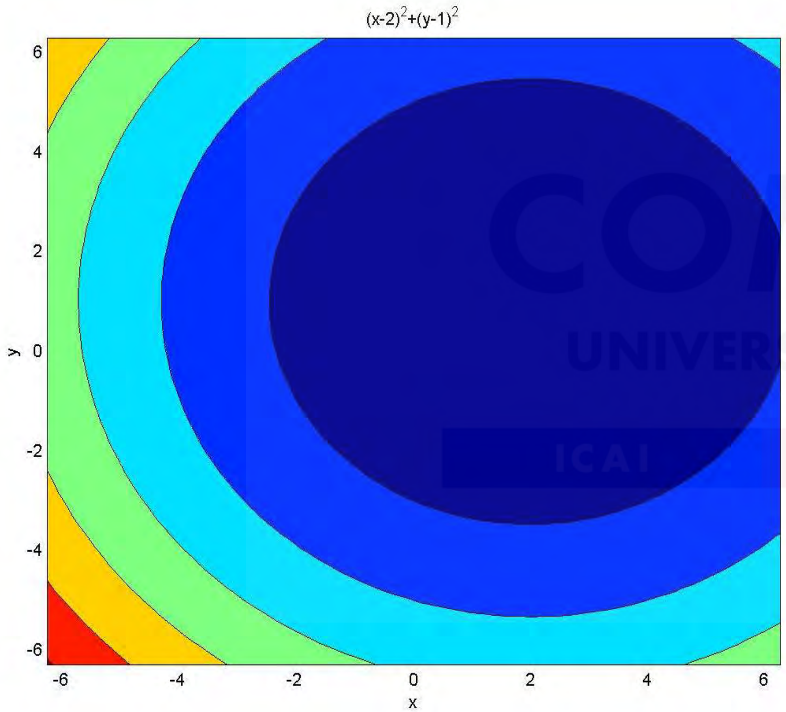


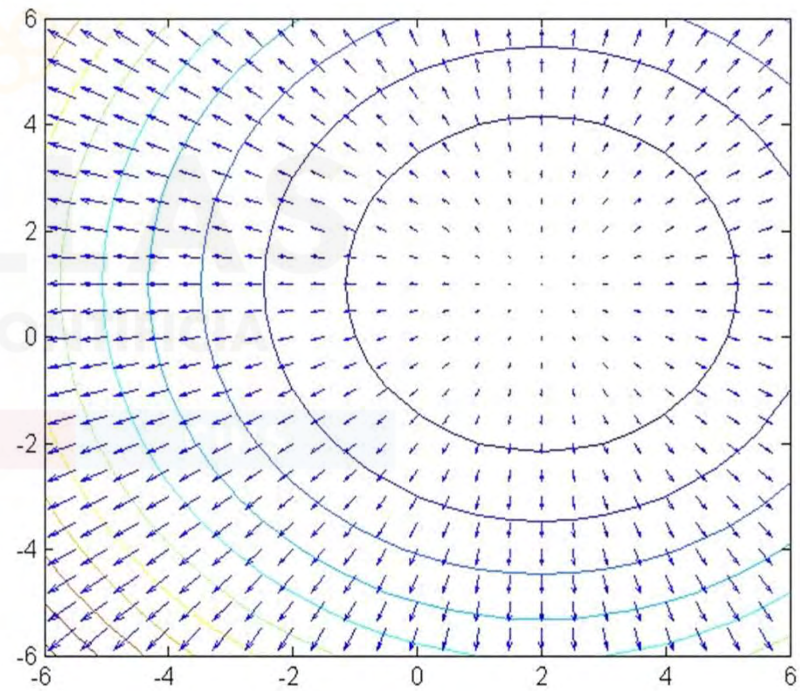
Example 3: Optimization WITHOUT constraints

$$f(x,y) = (x-2)^2 + (y-1)^2$$

```
ezcontourf('(x-2)^2+(y-1)^2')
```

```
[x,y] = meshgrid(-6:.5:6, -6:.5:6);
z=(x-2).^2+(y-1).^2;
[px,py] = gradient(z,0.5,0.5);
contour(x,y,z);
hold on, quiver(x,y,px,py)
```





Example 4: Optimization WITHOUT constraints

$$f(x,y) = 8x^2 + 3xy + 7y^2 - 25x + 31y - 29$$

- Gradient = 0 yields a system of equations

$$\nabla f(x,y) = \begin{pmatrix} 16x + 3y - 25 \\ 3x + 14y + 31 \end{pmatrix} = 0$$

with solution $(x,y) = (2.060, -2.656)$

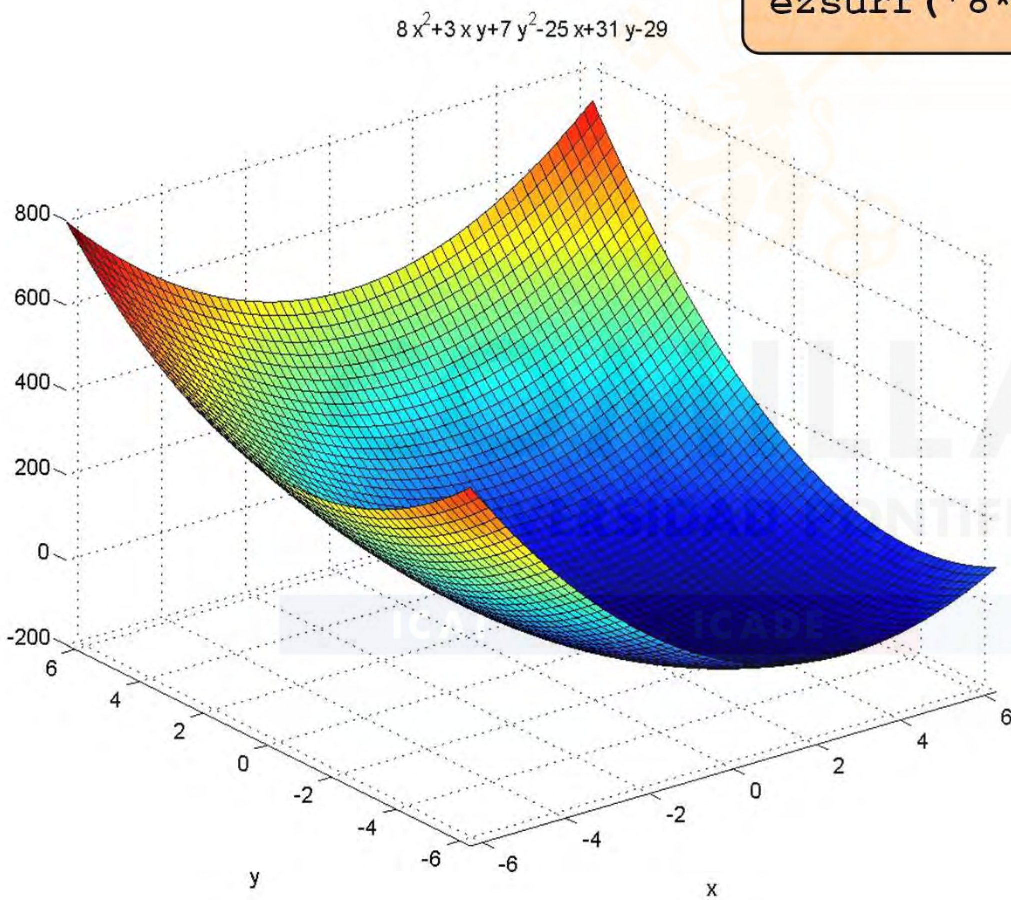
- Hessian $\nabla^2 f(x,y) = \begin{pmatrix} 16 & 3 \\ 3 & 14 \end{pmatrix}$ is a positive definite matrix

and (independent of the point) this yields a global minimum

Example 4: Optimization WITHOUT constraints

$$f(x,y) = 8x^2 + 3xy + 7y^2 - 25x + 31y - 29$$

```
ezsurf('8*x^2+3*x*y+7*y^2-25*x+31*y-29')
```

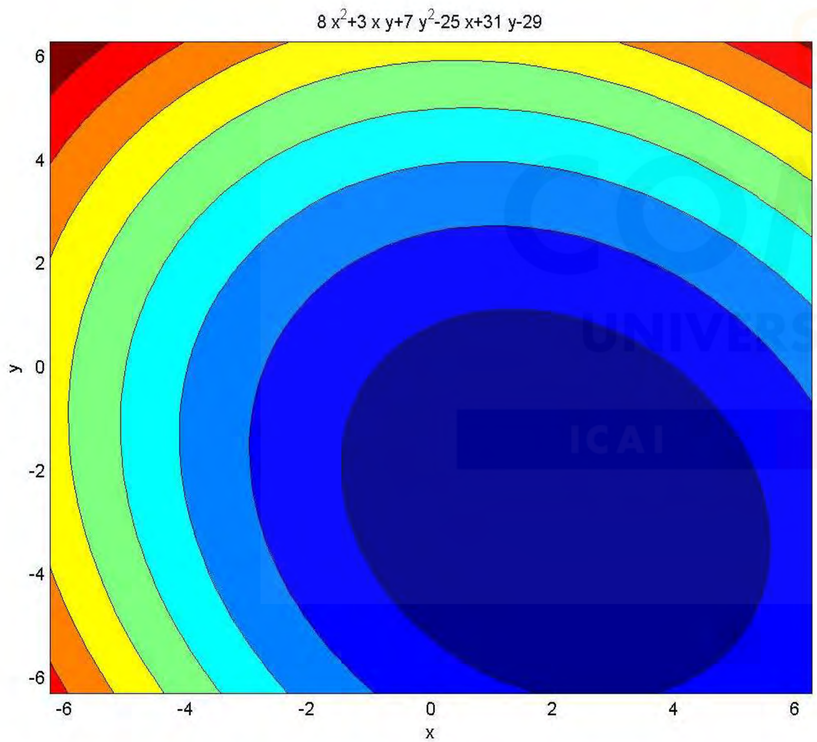


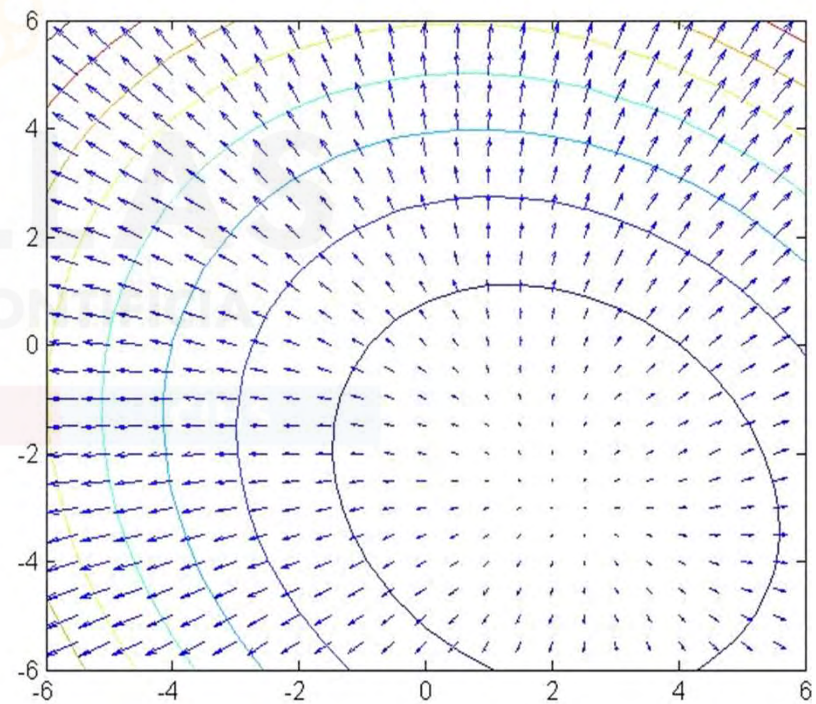
Example 4: Optimization WITHOUT constraints

$$f(x,y) = 8x^2 + 3xy + 7y^2 - 25x + 31y - 29$$

```
ezcontourf('8*x^2+3*x*y+7*y^2-25*x+31*y-29')
```

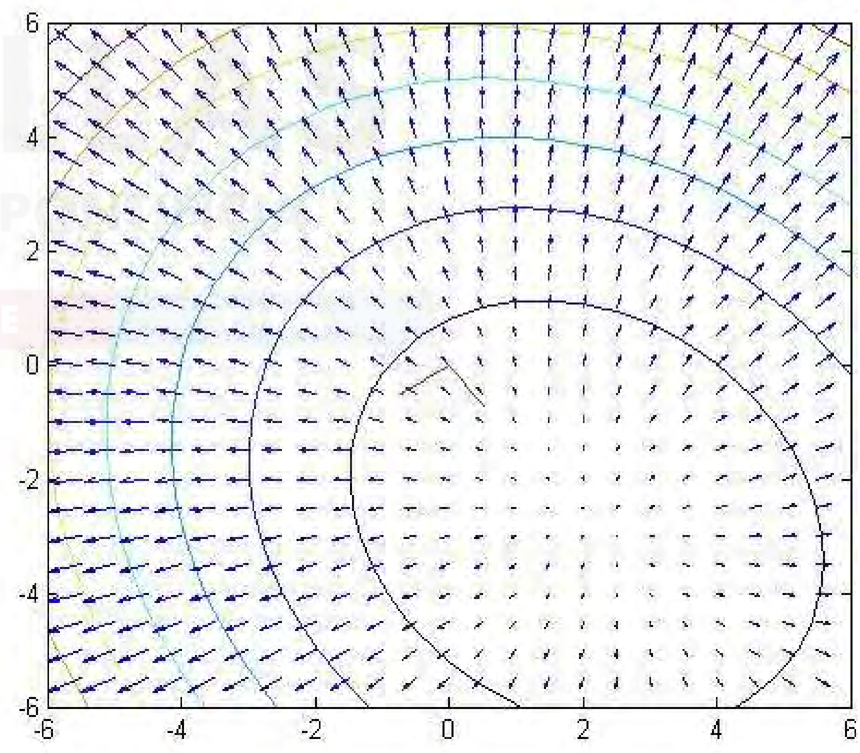
```
[x,y] = meshgrid(-6:.5:6, -6:.5:6);
z=8*x.^2+3*x.*y+7*y.^2-25*x+31*y-29;
[px,py] = gradient(z,0.5,0.5);
contour(x,y,z);
hold on, quiver(x,y,px,py)
```





Eigenvalues and eigenvectors

- Hessian $\nabla^2 f(x,y) = \begin{pmatrix} 16 & 3 \\ 3 & 14 \end{pmatrix}$ is a positive definite matrix
- Its eigenvalues are $\lambda = \begin{pmatrix} 11.84 \\ 18.16 \end{pmatrix}$
- Its eigenvectors are $v_1 = \begin{pmatrix} 0.58 \\ -0.81 \end{pmatrix}$ $v_2 = \begin{pmatrix} -0.81 \\ -0.58 \end{pmatrix}$



5

OPTIMALITY CONDITIONS FOR NLP

Lagrangian (i)

- Let's see this optimization problem $\min_x f(x)$ $Ax = b$

  being $x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$
- We define the Lagrangian as

$$L(x, \lambda) = f(x) + \lambda^T (Ax - b)$$

  being $\lambda \in \mathbb{R}^m$ the Lagrange multipliers.
- The Lagrangian problem is unconstrained.
- A constrained problem is transformed in an unconstrained one with m additional variables.
- The minimum of both problems coincides given that $Ax = b$

Lagrangian (ii)

- First order optimality conditions

$$\nabla L(x^*,\lambda^*) = 0 \quad \Rightarrow \quad \begin{cases} \nabla_x L(x^*,\lambda^*) = \nabla f(x^*) + A^T\lambda^* = 0 \\ \nabla_\lambda L(x^*,\lambda^*) = Ax^* - b = 0 \end{cases}$$

  and therefore if x^* is a local minimum must satisfy $\nabla f(x^*) = -A^T\lambda^*$

- In a local minimum the objective function gradient is a linear combination of the constraint gradients and the Lagrange multipliers are the weights.
- Multipliers represent the change in the objective function for a per unit change (marginal) in the bound of each constraint. In the particular case of LP those are named *dual variables* or *shadow prices*. With this formulation multipliers result with opposite sign to the dual variables.

Lagrangian (iii)

- Let's see this optimization problem
$$\min_x f(x)$$
$$g_i(x) \le 0 \quad i = 1,\dots,m$$
$$h_j(x) = 0 \quad j = 1,\dots,l$$

  being $f: \mathbb{R}^n \to \mathbb{R}$, $g_i: \mathbb{R}^n \to \mathbb{R}$
- Let's define the Lagrangian as

$$L(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{l} \mu_j h_j(x)$$

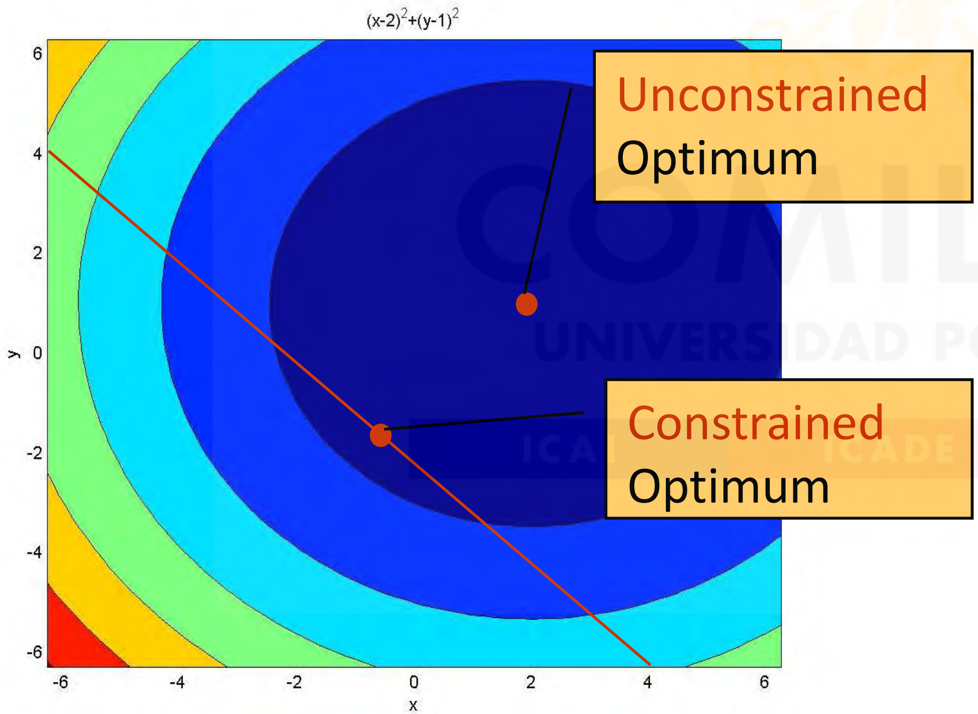
  where $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^l$ are the Lagrange multipliers.
- Lagrangian is always a lower bound of $f(x)$ for feasible values of x and known values of $\lambda \ge 0$ (nonnegative) and μ (free).

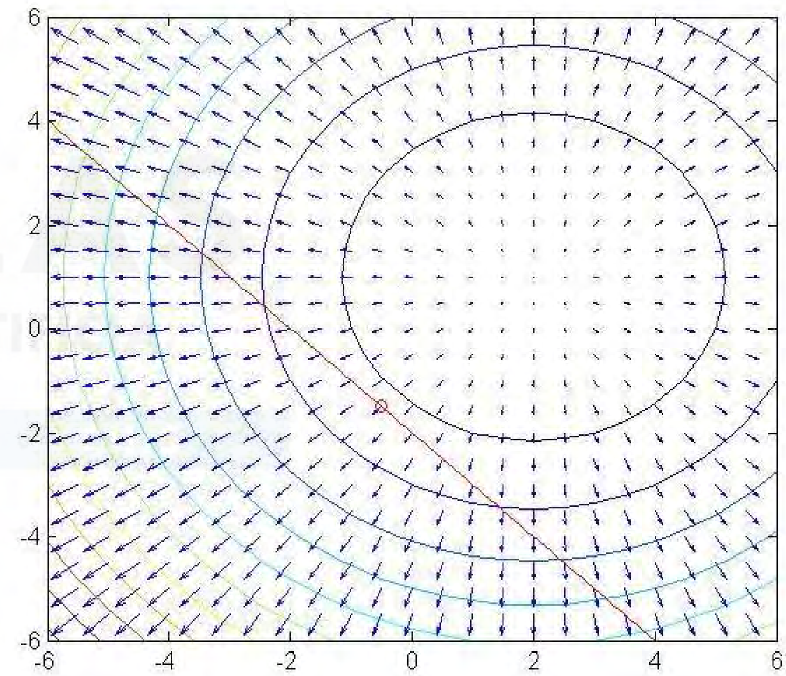
Example 1 (i)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y = -2$$

- Optimal point (-0.5,-1.5)
- Gradient $\nabla f(x^*, y^*) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix}_{(-0.5,-1.5)} = \begin{pmatrix} -5 \\ -5 \end{pmatrix} = -(1 \quad 1)^T \lambda^*$

$\lambda^* = 5$

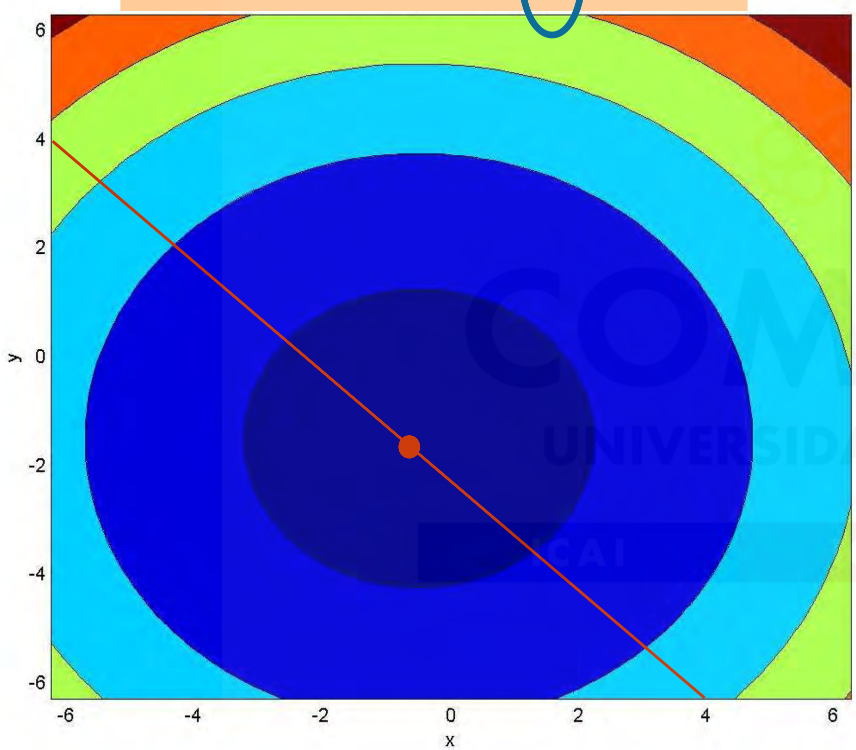




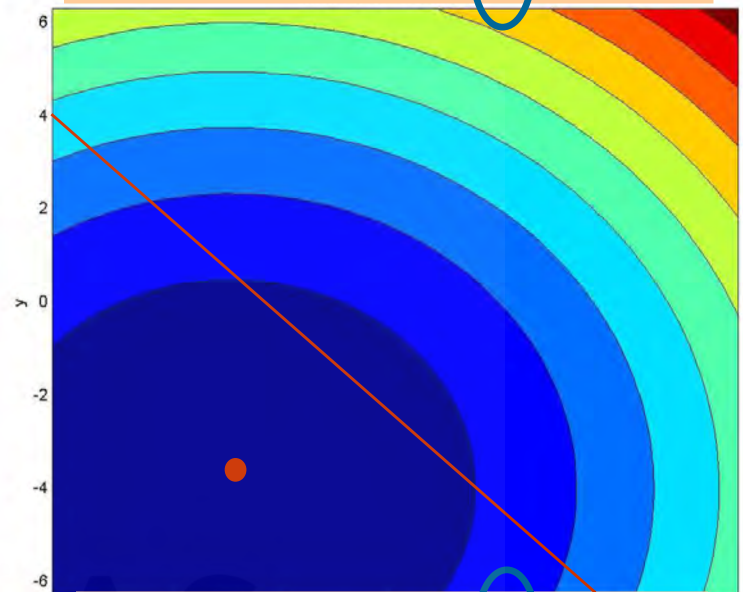
Example 1 (ii)

$\min(x-2)^2+(y-1)^2+\lambda(x+y+2)$

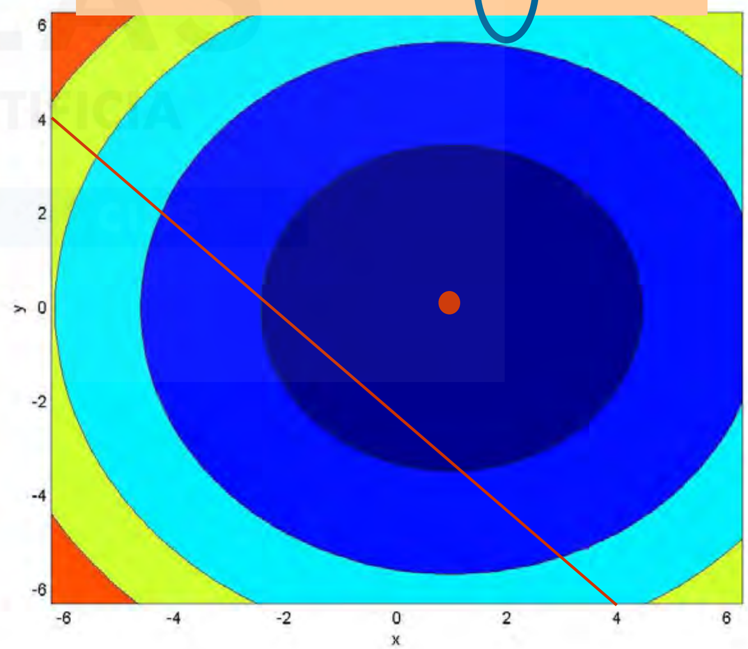
$\min(x-2)^2+(y-1)^2+5(x+y+2)$



$\min(x-2)^2+(y-1)^2+10(x+y+2)$

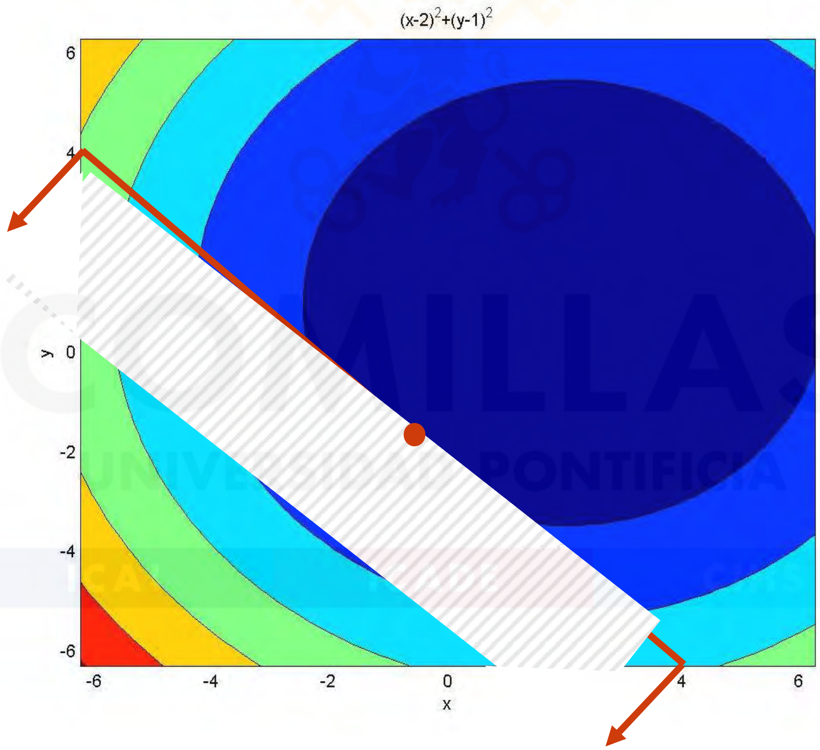


$\min(x-2)^2+(y-1)^2+2(x+y+2)$



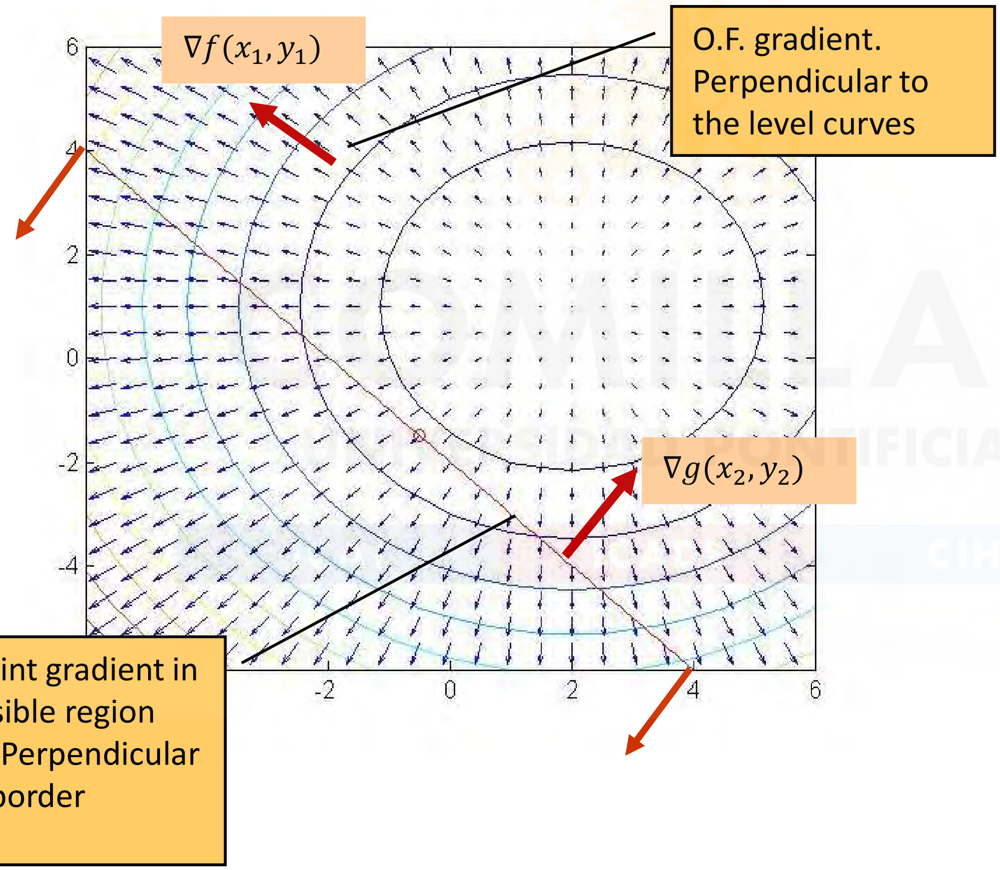
Example 2 (i)

$$\min(x-2)^2+(y-1)^2$$
$$x+y\leq -2$$



Example 2 (ii)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \le -2$$



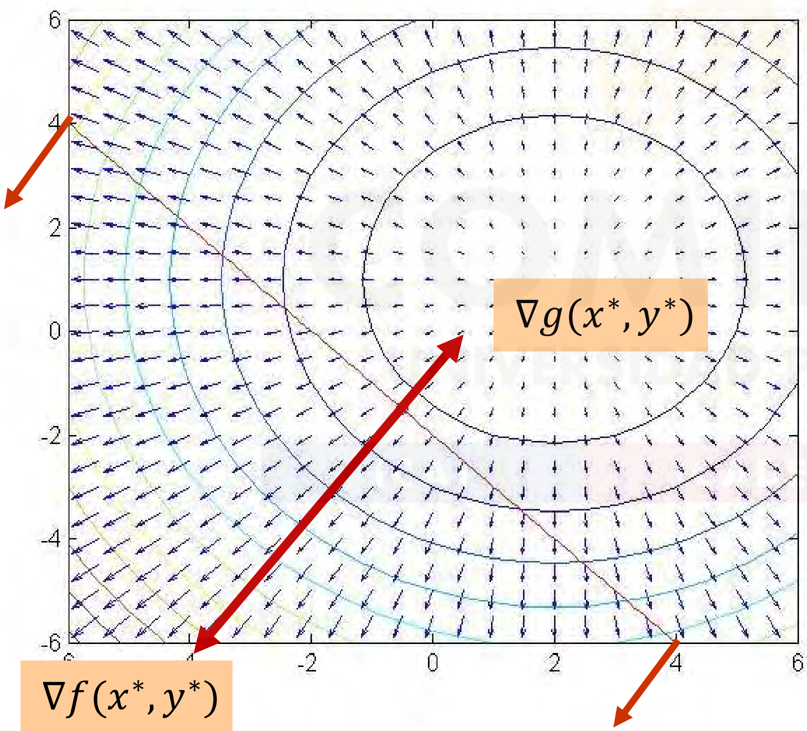
$$\nabla f(x,y) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix}$$

$$\nabla g(x,y) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Example 2 (iii)

$$\min(x-2)^2+(y-1)^2$$
$$x+y\le -2$$

- In the optimum (-0.5,-1.5) both gradients have opposite direction



$\nabla f(x^*,y^*)$

$$\nabla f(x^*,y^*)=\begin{pmatrix}2(x-2)\\2(y-1)\end{pmatrix}_{(-0.5,-1.5)}=\begin{pmatrix}-5\\-5\end{pmatrix}$$

$$-\nabla g(x^*,y^*)=\begin{pmatrix}-1\\-1\end{pmatrix}$$

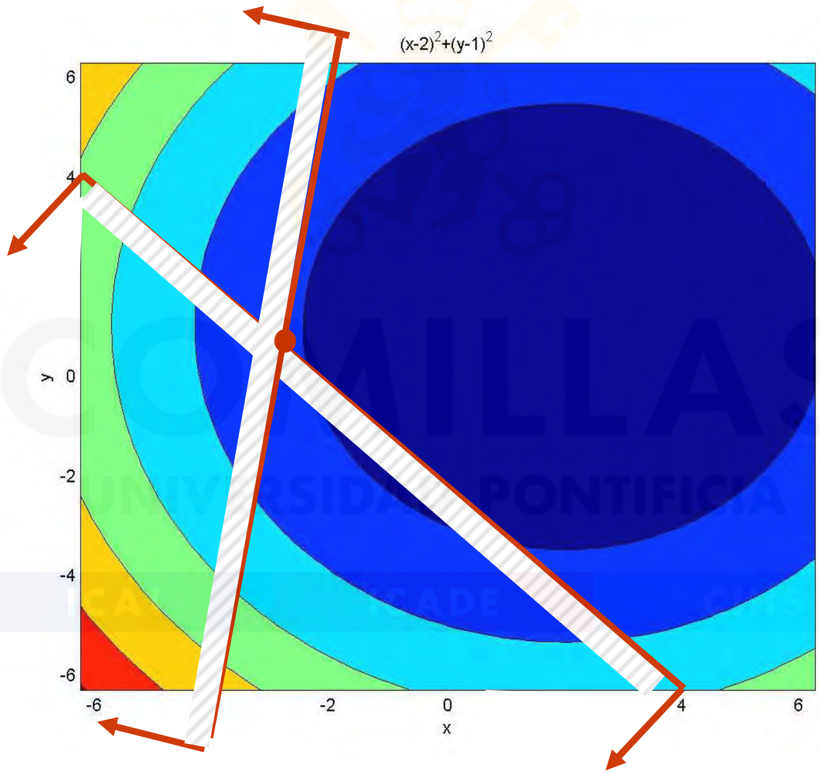
$$\nabla f(x^*,y^*)=-\lambda\nabla g(x^*,y^*)$$
$$\lambda\ge 0$$

$$\begin{pmatrix}-5\\-5\end{pmatrix}=\lambda\begin{pmatrix}-1\\-1\end{pmatrix}$$
$$\lambda\ge 0$$

$\lambda=5$

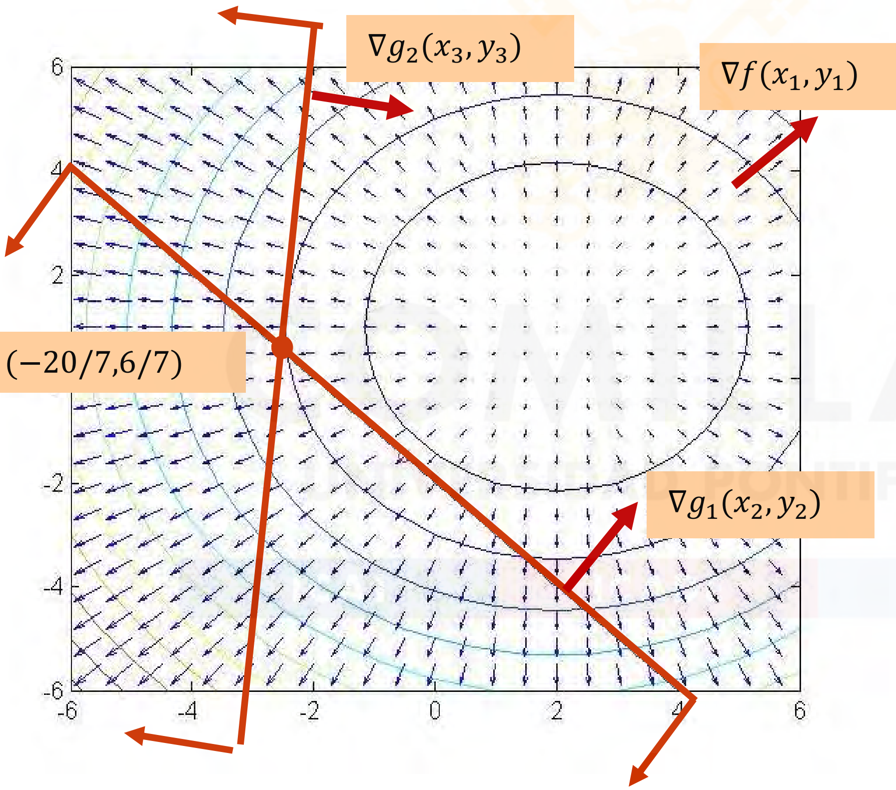
Example 3 (i)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \leq -2$$
$$6x - y \leq -18$$



Example 3 (ii)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \leq -2$$
$$6x - y \leq -18$$



$$\nabla f(x,y) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix}$$

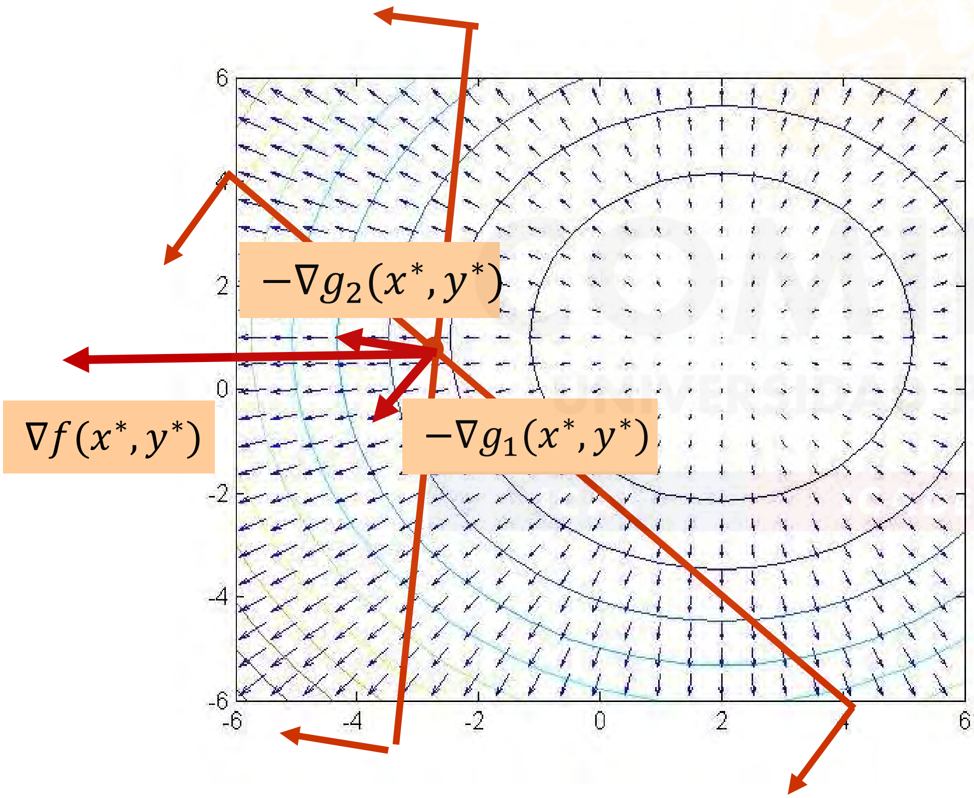
$$\nabla g_1(x,y) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\nabla g_2(x,y) = \begin{pmatrix} 6 \\ -1 \end{pmatrix}$$

Example 3 (iii)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \leq -2$$
$$6x - y \leq -18$$

- In the optimum (-20/7,6/7) the gradient of the o.f. can be expressed as a linear combination of the gradients of the constrains changed in sign



$$\nabla f(x^*, y^*) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix}_{(-20/7, 6/7)} = \begin{pmatrix} -68/7 \\ -2/7 \end{pmatrix}$$

$$-\nabla g_1(x^*, y^*) = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \qquad -\nabla g_2(x^*, y^*) = \begin{pmatrix} -6 \\ 1 \end{pmatrix}$$

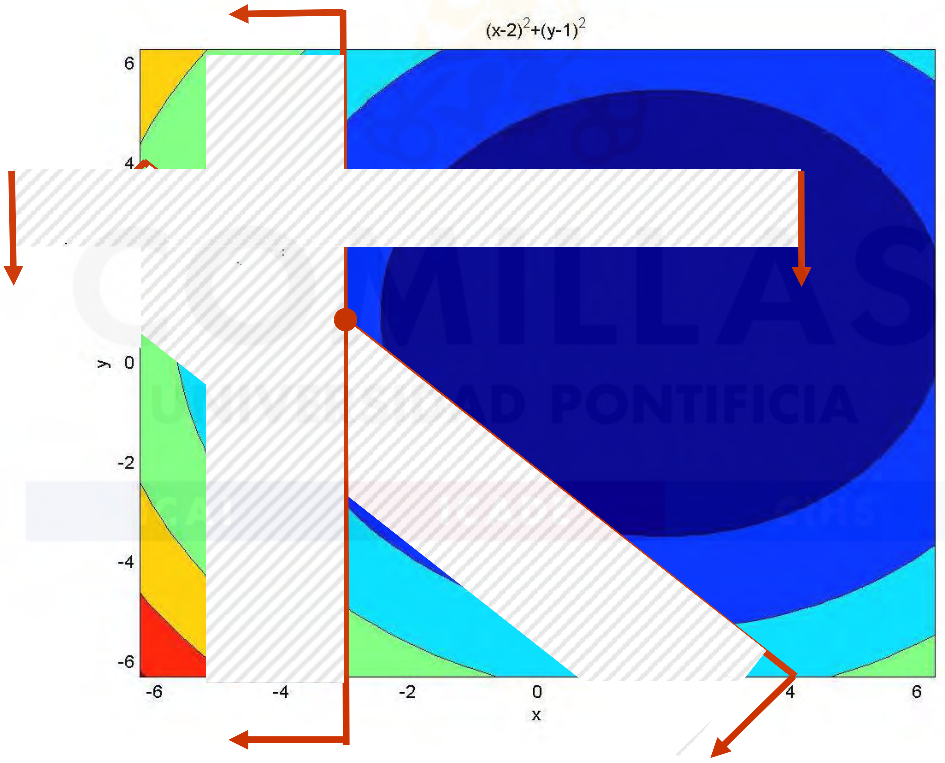
$$\nabla f(x^*, y^*) = -\lambda_1 \nabla g_1(x^*, y^*) - \lambda_2 \nabla g_2(x^*, y^*)$$
$$\lambda_1, \lambda_2 \geq 0$$

$$\begin{pmatrix} -68/7 \\ -2/7 \end{pmatrix} = \lambda_1 \begin{pmatrix} -1 \\ -1 \end{pmatrix} + \lambda_2 \begin{pmatrix} -6 \\ 1 \end{pmatrix}$$
$$\lambda_1, \lambda_2 \geq 0$$

$$\lambda_1 = 2.75$$
$$\lambda_2 = 1.125$$

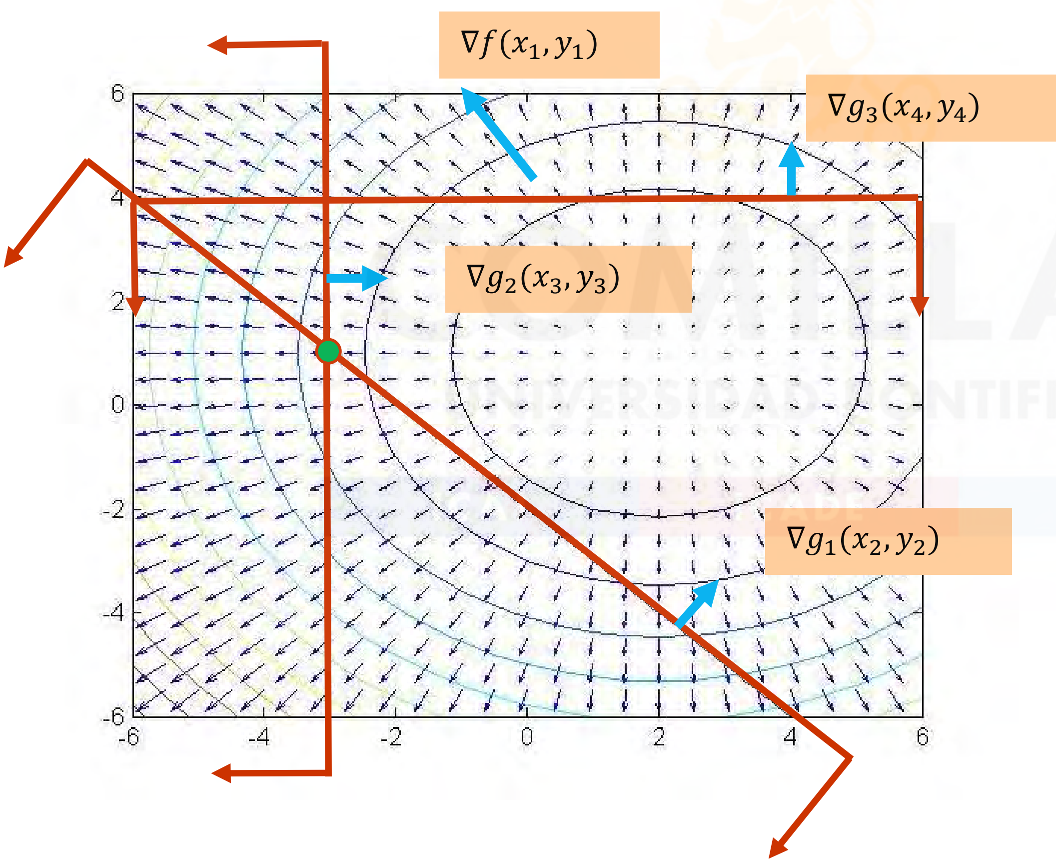
Example 4 (i)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \leq -2$$
$$x \leq -3$$
$$y \leq 4$$



Example 4 (ii)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \leq -2$$
$$x \leq -3$$
$$y \leq 4$$



$$\nabla f(x,y) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix}$$

$$\nabla g_1(x,y) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

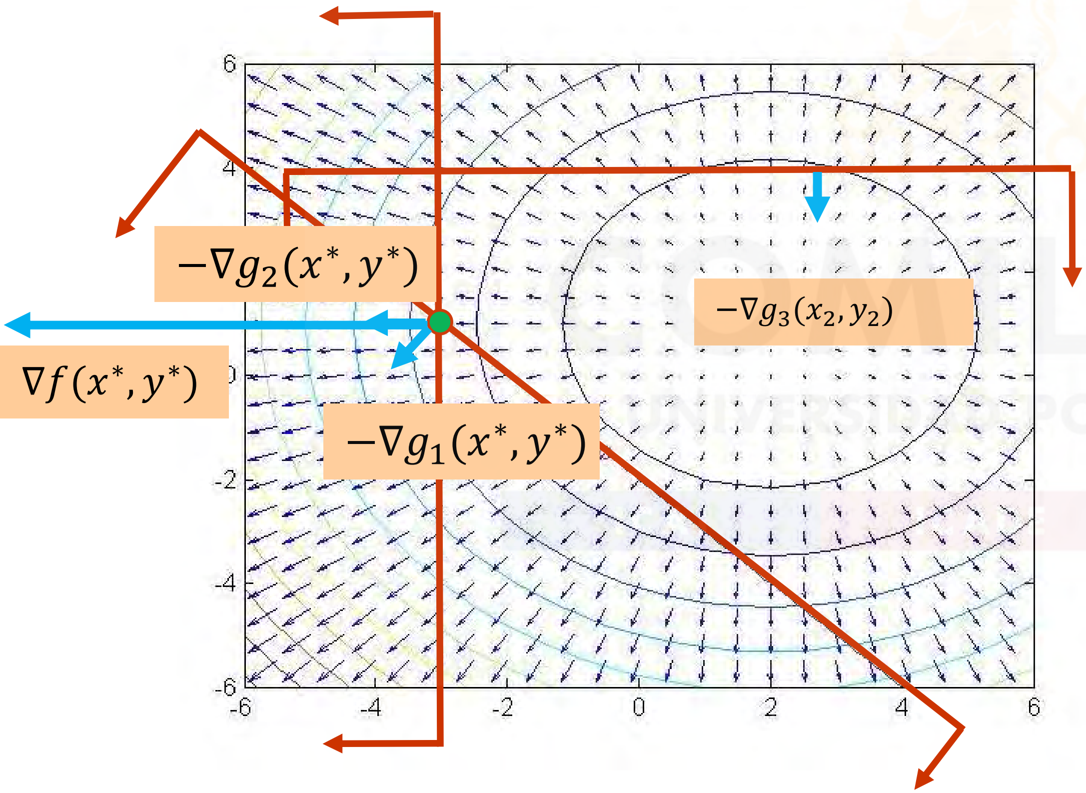
$$\nabla g_2(x,y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\nabla g_3(x,y) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Example 4 (iii)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y \leq -2$$
$$x \leq -3$$
$$y \leq 4$$

- In the optimum (-3,1) the gradient of the o.f. can be expressed as a linear combination of the gradients of the **binding constrains** changed in sign



$$\nabla f(x^*,y^*) = \begin{pmatrix} 2(x-2) \\ 2(y-1) \end{pmatrix}_{(-3,1)} = \begin{pmatrix} -10 \\ 0 \end{pmatrix}$$

$$-\nabla g_1(x^*,y^*) = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \qquad -\nabla g_2(x^*,y^*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$-\nabla g_3(x^*,y^*) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$\nabla f(x^*,y^*) = -\lambda_1 \nabla g_1(x^*,y^*) - \lambda_2 \nabla g_2(x^*,y^*)$$
$$\lambda_1, \lambda_2 \geq 0$$

$$\begin{pmatrix} -10 \\ 0 \end{pmatrix} = \lambda_1 \begin{pmatrix} -1 \\ -1 \end{pmatrix} + \lambda_2 \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$
$$\lambda_1, \lambda_2 \geq 0$$

$\lambda_1 = 0$
$\lambda_2 = 10$
$\lambda_3 = 0$

- ❑ The first constraint is superfluous. Degenerate solution.

Necessary conditions with inequality constraints (i)

- Let's see this optimization problem

$$\min_x f(x)$$
$$g_i(x) \leq 0 \quad i = 1, \dots, m$$

being $f: \mathbb{R}^n \to \mathbb{R}$, $g_i: \mathbb{R}^n \to \mathbb{R}$

- Let's be x^* a feasible point

$I = \{i / g_i(x^*) = 0\}$ the set of binding constraints

f and $\{g_i, i \in I\}$ differentiable in x^*

$\{g_i, i \notin I\}$ continuous in x^*

$\{\nabla g_i(x^*)\}_{i \in I}$ linearly independent

Necessary conditions with inequality constraints (ii)

$$\min_x f(x)$$
$$g_i(x) \le 0 \quad i = 1, \dots, m$$

- If x^* is a local minimum, then there exist an scalars $\{\lambda_i, i \in I\}$ such that

$$\nabla f(x^*) + \sum_{i \in I} \lambda_i^* \nabla g_i(x^*) = 0$$
$$\lambda_i^* \ge 0 \quad \forall i \in I$$

- Besides, if functions $\{g_i, i \notin I\}$ are differentiable in x^*, if x^* is a local optimum then

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0$$
$$\lambda_i^* g_i(x^*) = 0 \quad i = 1, \dots, m$$
$$\lambda_i^* \ge 0 \quad i = 1, \dots, m$$

Complementary Slackness Condition

- Nonbinding constraint → multiplier 0.
  Binding constraint → multiplier does not necessarily have to be 0.

Necessary conditions with inequality constraints (iii)

- Necessary conditions with inequality constraints

$$\min_x f(x)$$
$$g_i(x) \le 0 \quad i = 1, \ldots, m$$

- First order Karush-Kuhn-Tucker (KKT) necessary conditions for a local optimum

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0$$
$$\lambda_i^* g_i(x^*) = 0 \quad i = 1, \ldots, m$$
$$g_i(x^*) \le 0 \quad i = 1, \ldots, m$$
$$\lambda_i^* \ge 0 \quad i = 1, \ldots, m$$

Gradient of the o.f. is a linear combination of the gradients if the constraints changed in sign

Feasible point

Complementary Slackness Condition
Non binding constraint λ=0
Binding constraint $\lambda \neq 0$

Necessary conditions with inequality constraints (v)

- Let's see the problem

$$\min_x f(x)$$
$$g_i(x) \le 0 \quad i = 1, \dots, m$$

- The Lagrangian is

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$
$$\lambda_i \ge 0$$

- Optimality condition for the Lagrangian

$$\nabla L(x^*,\lambda^*) = 0 \quad \Rightarrow \quad \begin{cases} \nabla_x L(x^*,\lambda^*) = \nabla f(x^*) + \sum_{i \in I} \lambda_i^* \nabla g_i(x^*) = 0 \\ \nabla_\lambda L(x^*,\lambda^*) = g_i(x^*) = 0 \quad \forall i \in I \end{cases}$$

the 2nd one corresponds to the definition of binding constraints $\forall i \in I$

- To consider all the constraints is expressed as

$$\lambda_i^* g_i(x^*) = 0 \quad i = 1, \dots, m$$

Sufficient conditions with equality constraints (i)

- If the o.f. is not convex or the feasible region is not convex there still can be points that satisfy the necessary conditions

- Let x^* be a feasible point

  $I = \{i / g_i(x^*) = 0\}$ is the set of binding constraints

  f and $\{g_i, i \in I\}$ are convex and differentiable in **all of the feasible region**

- If there exist scalars $\{\lambda_i, i \in I\}$ such that

$$\nabla f(x^*) + \sum_{i \in I} \lambda_i \nabla g_i(x^*) = 0$$
$$\lambda_i \geq 0 \quad \forall i \in I$$

then x^* is a global minimum

Sufficient conditions with equality constraints (ii)

- Condition for a strict local minimum

  Alternatively, instead of the conditions that f and $\{g_i, i \in I\}$ are convex and differentiable in x^*, it can also be expressed as the condition that the Lagrangian $L(x) = f(x) + \sum_{i \in I} \lambda_i^* g_i(x)$,

  where λ_i^* represent the Lagrange multipliers of the constraints, must have a Hessian $\nabla^2 L(x^*) = \nabla^2 f(x^*) + \sum_{i \in I} \lambda_i^* \nabla^2 g_i(x^*)$ which is a positive definite matrix in x^*.

- The sufficient conditions for the case of maximization can be translated into the conditions that f must be concave in the point, that the constraints do not change and that the multipliers are smaller or equal to 0.

Necessary conditions with equality and inequality constraints (i)

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) \leq 0 \quad i = 1,\ldots,m$$
$$h_j(x) = 0 \quad j = 1,\ldots,l$$

where $f: \mathbb{R}^n \to \mathbb{R}$ $\quad g_i: \mathbb{R}^n \to \mathbb{R}$

- Let x^* be a feasible point

$I = \{i / g_i(x^*) = 0\}$ is the set of binding constraints

f and $\{g_i, i \in I\}$ are differentiable in x^*

$\{g_i, i \notin I\}$ are continuous in x^*

$\{h_j, j = 1,\ldots,l\}$ are continuously differentiable in x^*

$\{\nabla g_i(x^*), i \in I; \nabla h_j(x^*), j = 1,\ldots,l\}$ are linearly independent

Necessary conditions with equality and inequality constraints (ii)

- If x^* is a local minimum, then there exist scalars $\{\lambda_i, i \in I; \mu_j, j = 1, \dots, l\}$ such that

$$\nabla f(x^*) + \sum_{i \in I} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla h_j(x^*) = 0$$
$$\lambda_i \geq 0 \quad \forall i \in I$$

- Moreover, if the functions $\{g_i, i \notin I\}$ are differentiable in x^*, if x^* is a local optimum then

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla h_j(x^*) = 0$$
$$\lambda_i g_i(x^*) = 0 \quad i = 1, \dots, m$$
$$\lambda_i \geq 0 \quad i = 1, \dots, m$$

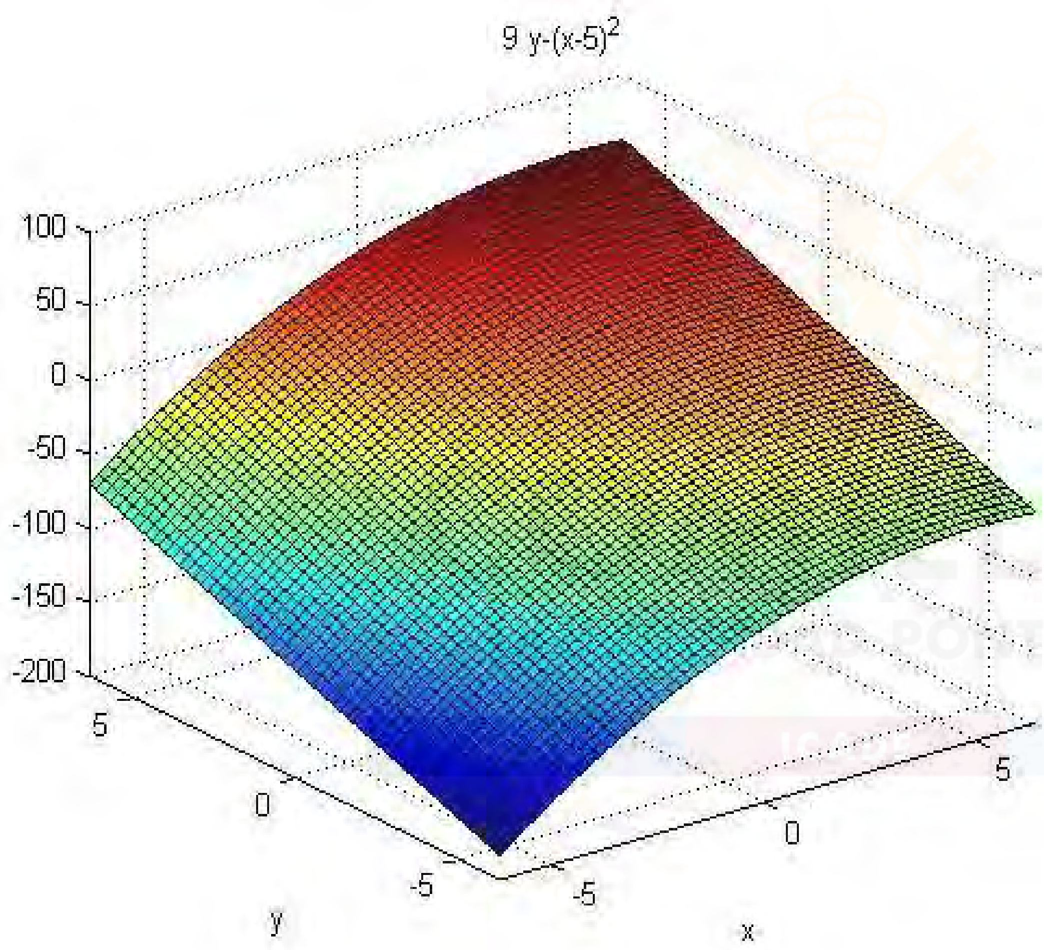
Example 5 (i)

$$\min_{x,y} f(x,y) = 9y - (x-5)^2$$

$$-x^2 + y \leq 0$$

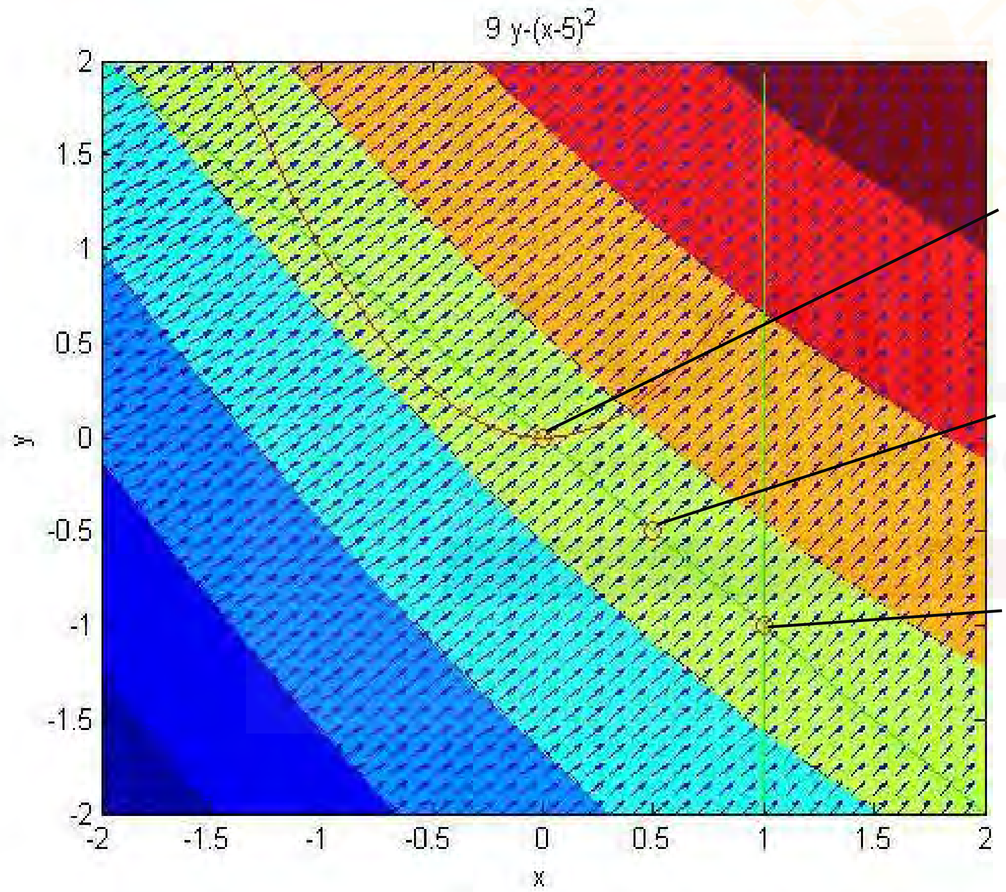
$$-x - y \leq 0$$

$$x - 1 \leq 0$$



Example 5 (ii)

$$\min_{x,y} f(x,y) = 9y - (x-5)^2$$
$$-x^2 + y \leq 0$$
$$-x - y \leq 0$$
$$x - 1 \leq 0$$



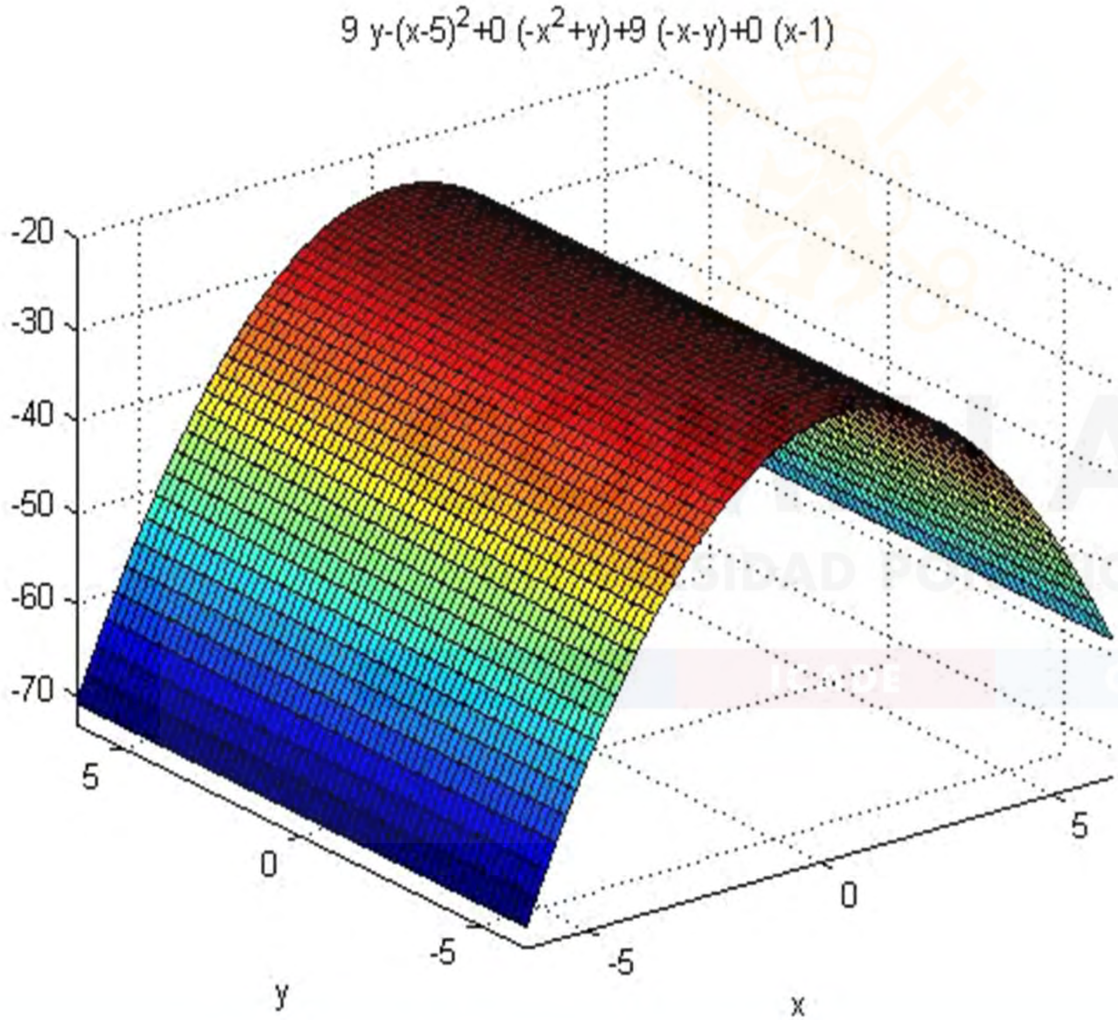
Point C (0,0,1,10,0) o.f.=-25

Point A (1/2,-1/2,0,9,0) o.f.=-24.75

Point B (1,-1,0,9,1) o.f.=-25

Example 5 (iii)

$$\min_{x,y,\lambda_1,\lambda_2,\lambda_3} L(x,y,\lambda_1,\lambda_2,\lambda_3) = 9y - (x-5)^2 + \lambda_1(-x^2+y) + \lambda_2(-x-y) + \lambda_3(x-1)$$



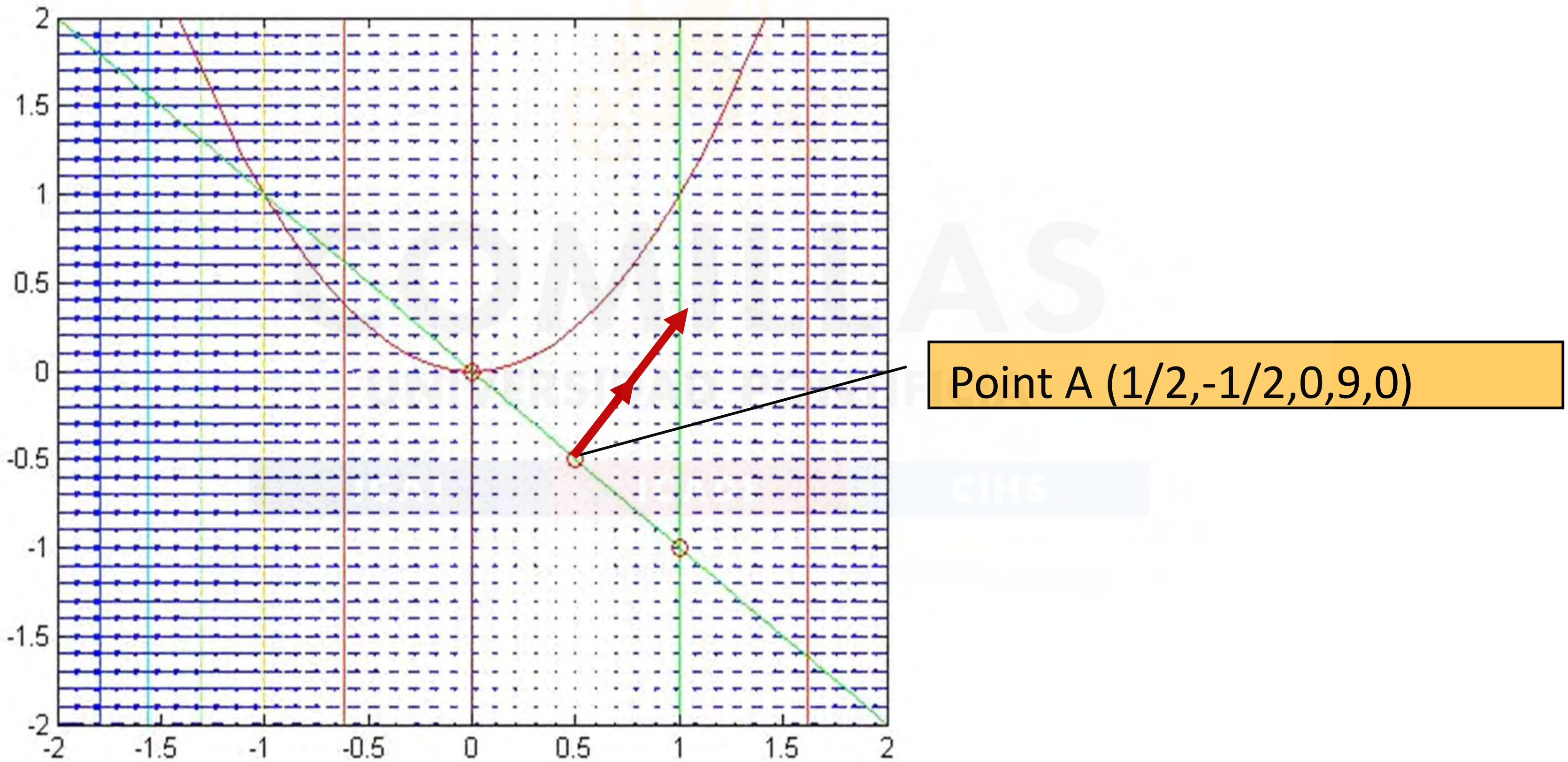
- Lagrangian for the multiplier values corresponding to point A (1/2,-1/2,0,9,0)

Unbounded function

Example 5 (iv)

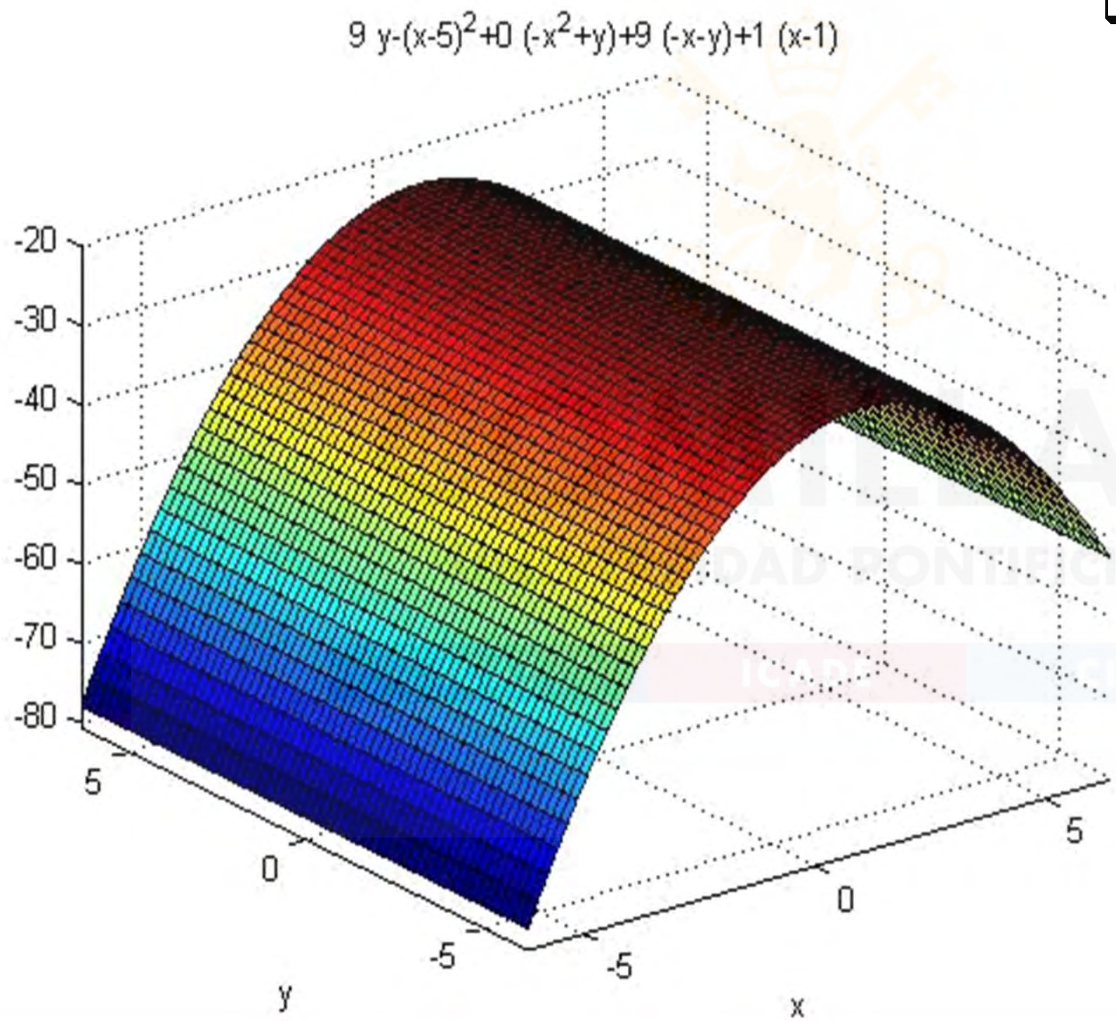
$$\min_{x,y,\lambda_1,\lambda_2,\lambda_3} L(x,y,\lambda_1,\lambda_2,\lambda_3) = 9y-(x-5)^2+\lambda_1(-x^2+y)+\lambda_2(-x-y)+\lambda_3(x-1)$$

- Lagrangian for multiplier values corresponding to point A (1/2,-1/2,0,9,0)



Example 5 (v)

$$\min_{x,y,\lambda_1,\lambda_2,\lambda_3} L(x,y,\lambda_1,\lambda_2,\lambda_3) = 9y - (x-5)^2 + \lambda_1(-x^2+y) + \lambda_2(-x-y) + \lambda_3(x-1)$$



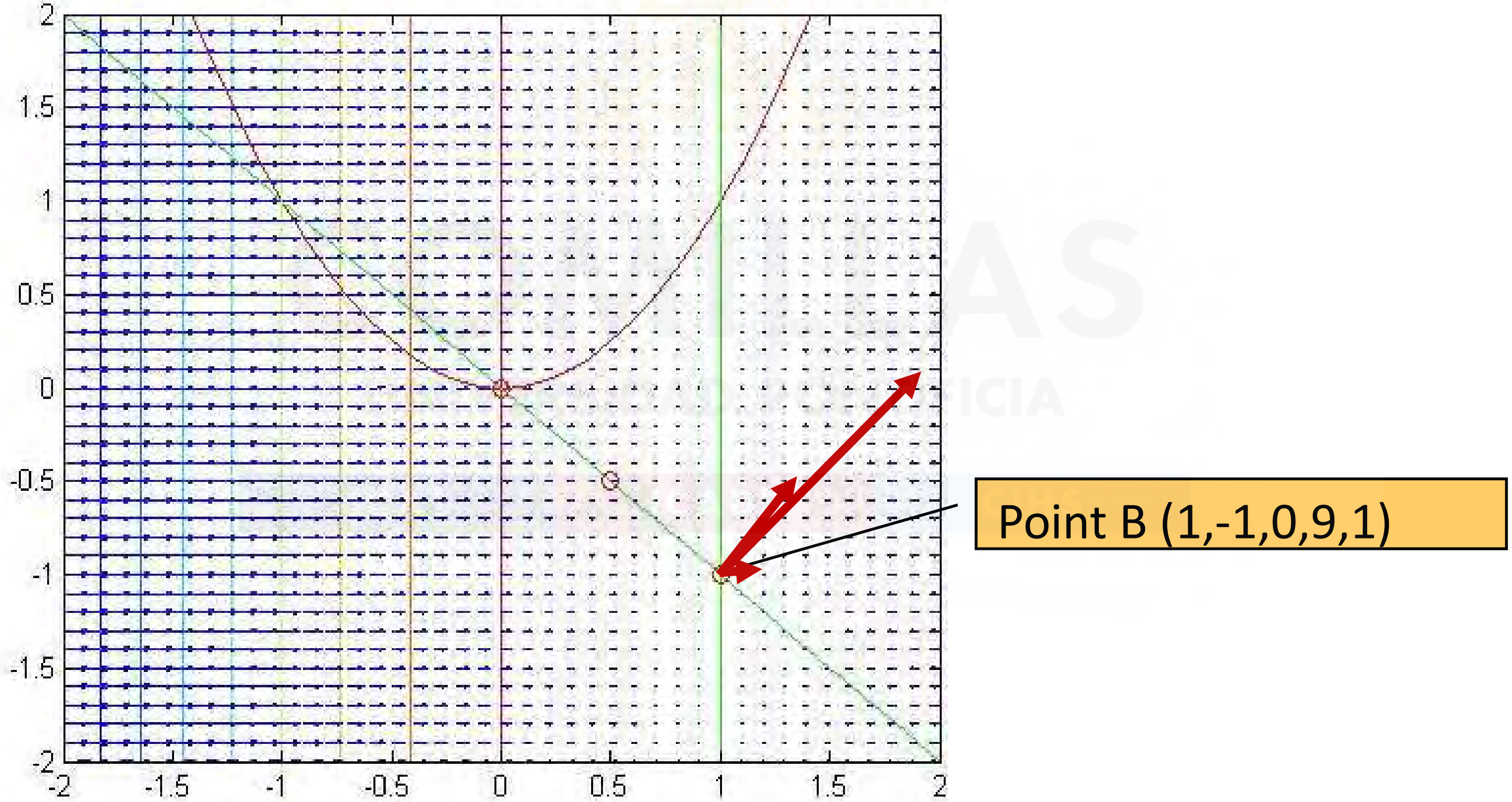
- Lagrangian for the multiplier values corresponding to point B (1,-1,0,9,1)

Unbounded function

Example 5 (vi)

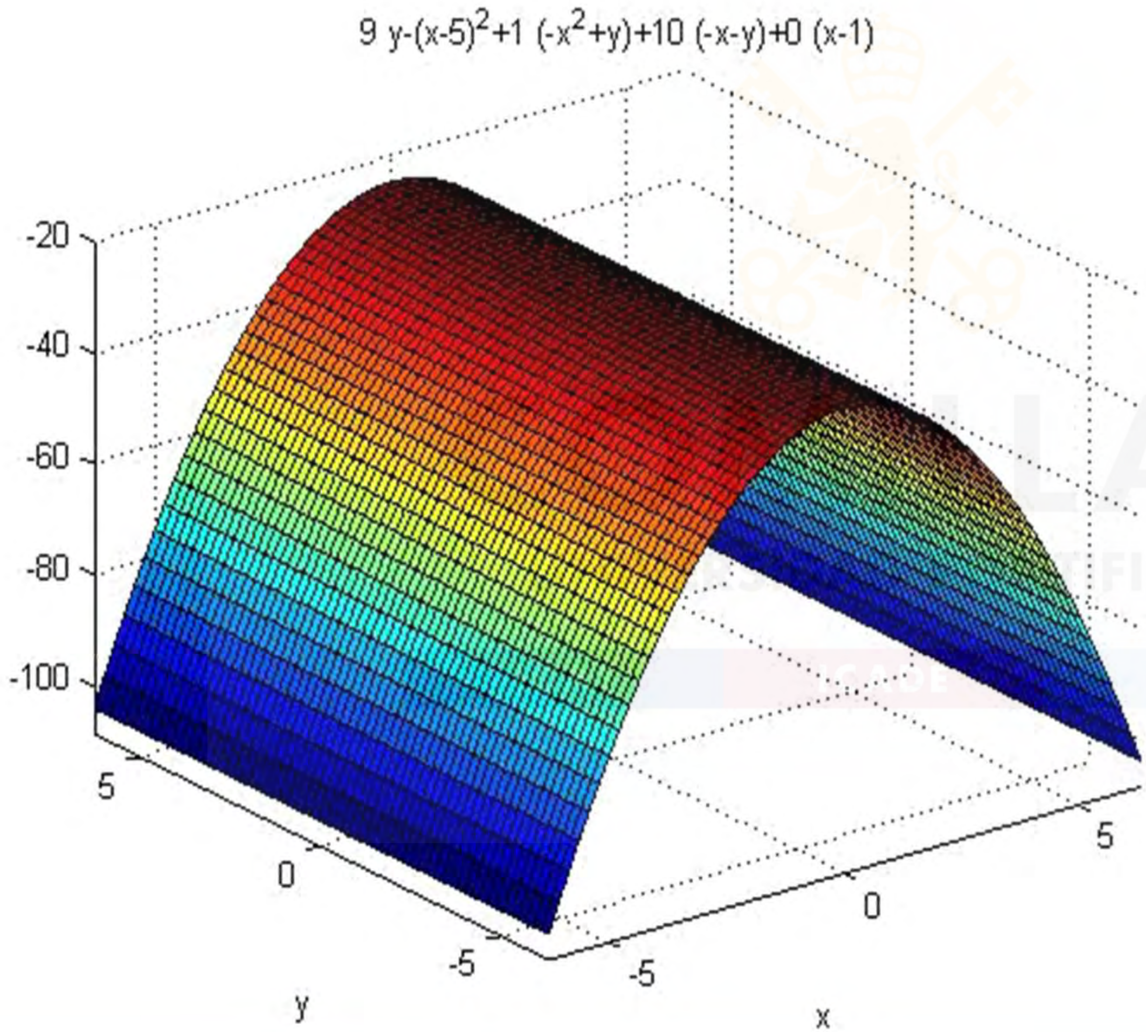
$$\min_{x,y,\lambda_1,\lambda_2,\lambda_3} L(x,y,\lambda_1,\lambda_2,\lambda_3) = 9y - (x-5)^2 + \lambda_1(-x^2 + y) + \lambda_2(-x - y) + \lambda_3(x-1)$$

- Lagrangian for multiplier values corresponding to point B (1,-1,0,9,1)



Example 5 (vii)

$$\min_{x,y,\lambda_1,\lambda_2,\lambda_3} L(x,y,\lambda_1,\lambda_2,\lambda_3) = 9y - (x-5)^2 + \lambda_1(-x^2 + y) + \lambda_2(-x-y) + \lambda_3(x-1)$$



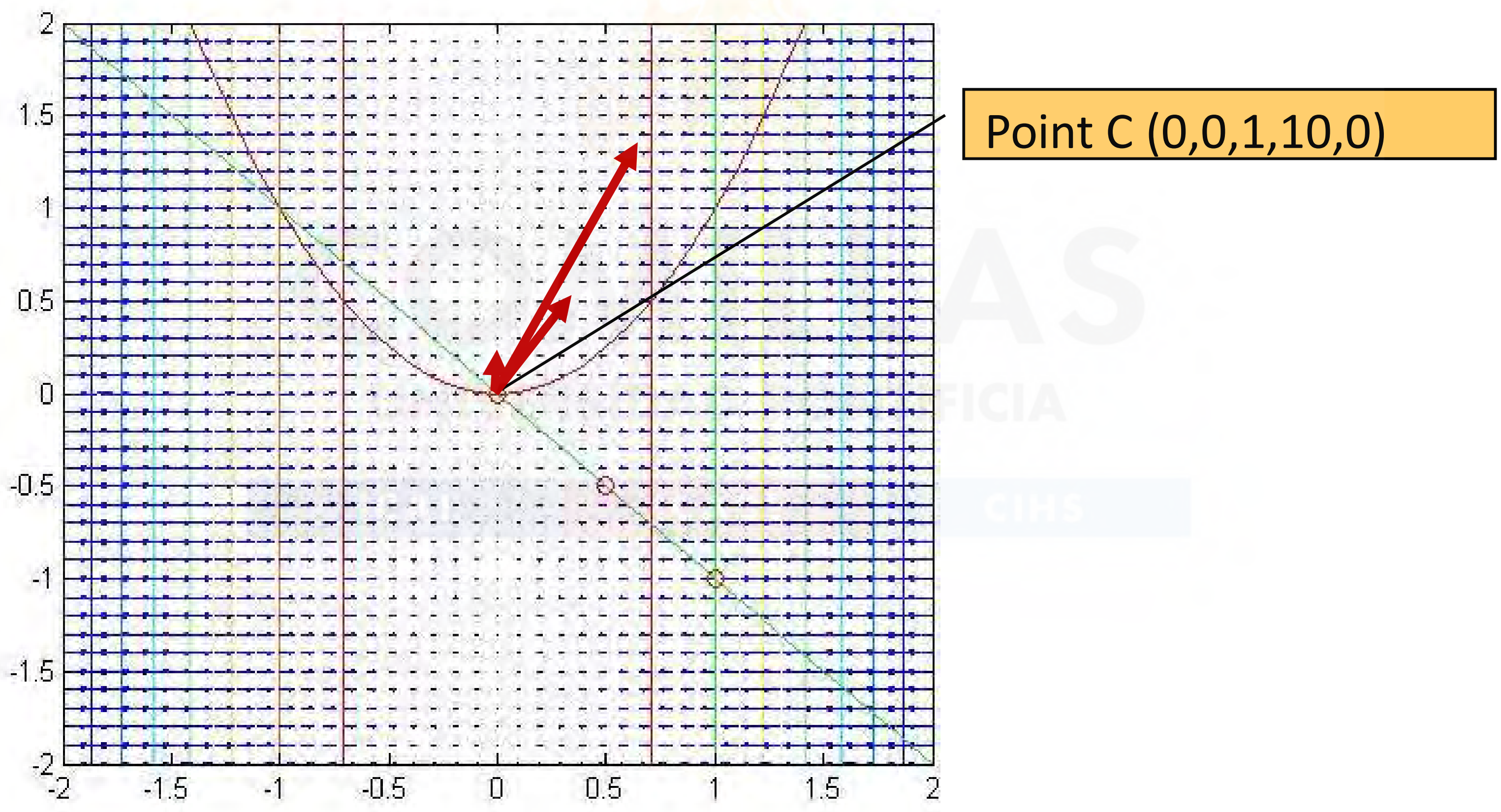
- Lagrangian for the multiplier values corresponding to point C (0,0,1,10,0)

Unbounded function

Example 5 (viii)

$$\min_{x,y,\lambda_1,\lambda_2,\lambda_3} L(x,y,\lambda_1,\lambda_2,\lambda_3) = 9y - (x-5)^2 + \lambda_1(-x^2+y) + \lambda_2(-x-y) + \lambda_3(x-1)$$

- Lagrangian for multiplier values corresponding to point C (0,0,1,10,0)



Necessary conditions with equality and inequality constraints (iii)

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) \leq 0 \quad i = 1,\ldots,m$$
$$h_j(x) = 0 \quad j = 1,\ldots,l$$

- The necessary first order Karush-Kuhn-Tucker (KKT) conditions for a local optimum

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{l} \mu_j^* \nabla h_j(x^*) = 0$$
$$\lambda_i^* g_i(x^*) = 0 \quad i = 1,\ldots,m$$
$$g_i(x^*) \leq 0 \quad i = 1,\ldots,m$$
$$h_j(x^*) = 0 \quad j = 1,\ldots,l$$
$$\lambda_i^* \geq 0 \quad i = 1,\ldots,m$$

Gradient of o.f.: Linear combination of the gradients of the constraints with changed sign

Complementary slackness conditions
Nonbinding constraint λ=0
Binding constraint $\lambda \neq 0$

Feasible point

Sufficient conditions with equality and inequality constraints (i)

- Let x^* be a feasible point

  $I = \{i / g_i(x^*) = 0\}$ is the set of binding constraints

  f and $\{g_i, i \in I\}$ are convex and differentiable in **all the feasible region**

- If there exist scalars $\{\lambda_i, i \in I; \mu_j, j = 1, \dots, l\}$ such that

$$\nabla f(x^*) + \sum_{i \in I} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla h_j(x^*) = 0$$
$$\lambda_i \geq 0 \quad \forall i \in I$$

so that h_j is convex in all the feasible region if $\mu_j > 0$ and h_j is concave in all the feasible region if $\mu_j < 0$, then x^* is a global minimum

Transportation problem solved as MCP

```
sets
   I origins      / VIGO, ALGECIRAS /
   J destinations / MADRID, BARCELONA, VALENCIA /

parameters
   pA(i) origin capacity
       / VIGO      350
         ALGECIRAS 700 /

   pB(j) destination demand
       / MADRID    400
         BARCELONA 450
         VALENCIA  150 /

table pC(i,j) per unit transportation cost
            MADRID BARCELONA VALENCIA
VIGO         0.06      0.12      0.09
ALGECIRAS    0.05      0.15      0.11

variables
   vX(i,j) units transported
   vCost   transportation cost

positive variable vX

equations
   eCost         transportation cost
   eCapacity(i) maximum capacity of each origin
   eDemand   (j) demand supply at destination ;

eCost        .. sum[(i,j), pC(i,j) * vX(i,j)] =e= vCost ;
eCapacity(i) .. sum[   j ,           vX(i,j)] =l= pA(i) ;
eDemand   (j) .. sum[ i   ,          vX(i,j)] =g= pB(j) ;

model mTransport / all /
solve mTransport using LP minimizing vCost
```

$$\min_x \sum_{ij} c_{ij} x_{ij}$$

$$\sum_j x_{ij} \le a_i \quad \forall i$$

$$\sum_i x_{ij} \ge b_j \quad \forall j$$

$$x_{ij} \ge 0$$

$$\mathcal{L} = \sum_{ij} c_{ij} x_{ij} + \alpha_i \left(\sum_j x_{ij} - a_i \right) + \beta_j \left(b_j - \sum_i x_{ij} \right)$$

$$\frac{\partial L}{\partial x_{ij}} \rightarrow$$

$$c_{ij} + \alpha_i \ge \beta_j \quad : x_{ij} \quad \forall ij$$

$$-\sum_j x_{ij} \ge -a_i \quad : \alpha_i \quad \forall i$$

$$\sum_i x_{ij} \ge b_j \quad : \beta_j \quad \forall j$$

$$x_{ij}, \alpha_i, \beta_j \ge 0$$

```
sets
   I origins      / VIGO, ALGECIRAS /
   J destinations / MADRID, BARCELONA, VALENCIA /

parameters
   pA(i) origin capacity
       / VIGO      350
         ALGECIRAS 700 /

   pB(j) destination demand
       / MADRID    400
         BARCELONA 450
         VALENCIA  150 /

table pC(i,j) per unit transportation cost
            MADRID BARCELONA VALENCIA
VIGO         0.06      0.12      0.09
ALGECIRAS    0.05      0.15      0.11

variables
   vX(i,j) units transported
   vA(i  ) Lagrange multiplier of capacity constraint
   vB(  j) Lagrange multiplier of demand   constraint

positive variables vX, vA, vB

equations
   eProfit(i,j) marginal cost >= marginal profit
   eCapacity(i) maximum capacity of each origin
   eDemand   (j) demand supply at destination ;

eProfit(i,j) ..  vA(i) + pC(i,j)  =g=  vB(j) ;
eCapacity(i) .. -sum[j,  vX(i,j)] =g= -pA(i) ;
eDemand   (j) ..  sum[i,  vX(i,j)] =g=  pB(j) ;

model mTransport / eProfit.vX eCapacity.vA eDemand.vB /
solve mTransport using MCP
```

6

METHODS FOR UNCONSTRAINED OPTIMIZATION (master)

COMILLAS UNIVERSIDAD PONTIFICIA ICAI ICADE CIHS

Classification of optimization methods WITHOUT constraints according to the use of derivatives

- Without derivatives
  - Random search method
  - Pattern search (Hooke-Jeeves method)
  - Rosenbrock method (of rotating or cyclic coordinates)
  - Nelder-Mead method
- First derivatives (gradient)
  - Method of steepest descent
  - Conjugate gradient method (Fletcher y Reeves)
- Second derivatives (Hessian)
  - Newton Method
  - Quasi Newton methods (Broyden-Fletcher-Goldfarb-Shanno BFGS, Davidon-Fletcher-Powell DFP)

Newton method for a one-dimensional function (i)

- The interpolation algorithms carry out an approximation of the function f in each iteration, in the point x_k considered in this iteration, by a polynomial of second or third degree.
- This method fits, in iteration k, a parabola $q(x)$ to $f(x)$ and takes x_{k+1} as the vertex of this parabola.

$$q(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + \frac{1}{2} f''(x_k)(x_{k+1} - x_k)^2$$

$$q'(x_{k+1}) = 0 \Rightarrow x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

- The algorithm has quadratic convergence under certain conditions, but it is very unstable and usually it is necessary to take precautions and to include protections.

Newton method for a one-dimensional function (ii)

$f(x) = (x-1)^3 + 2(x-1)^2 + 3$ $\quad f'(x) = 3(x-1)^2 + 4(x-1)$ $\quad f''(x) = 6(x-1) + 4$

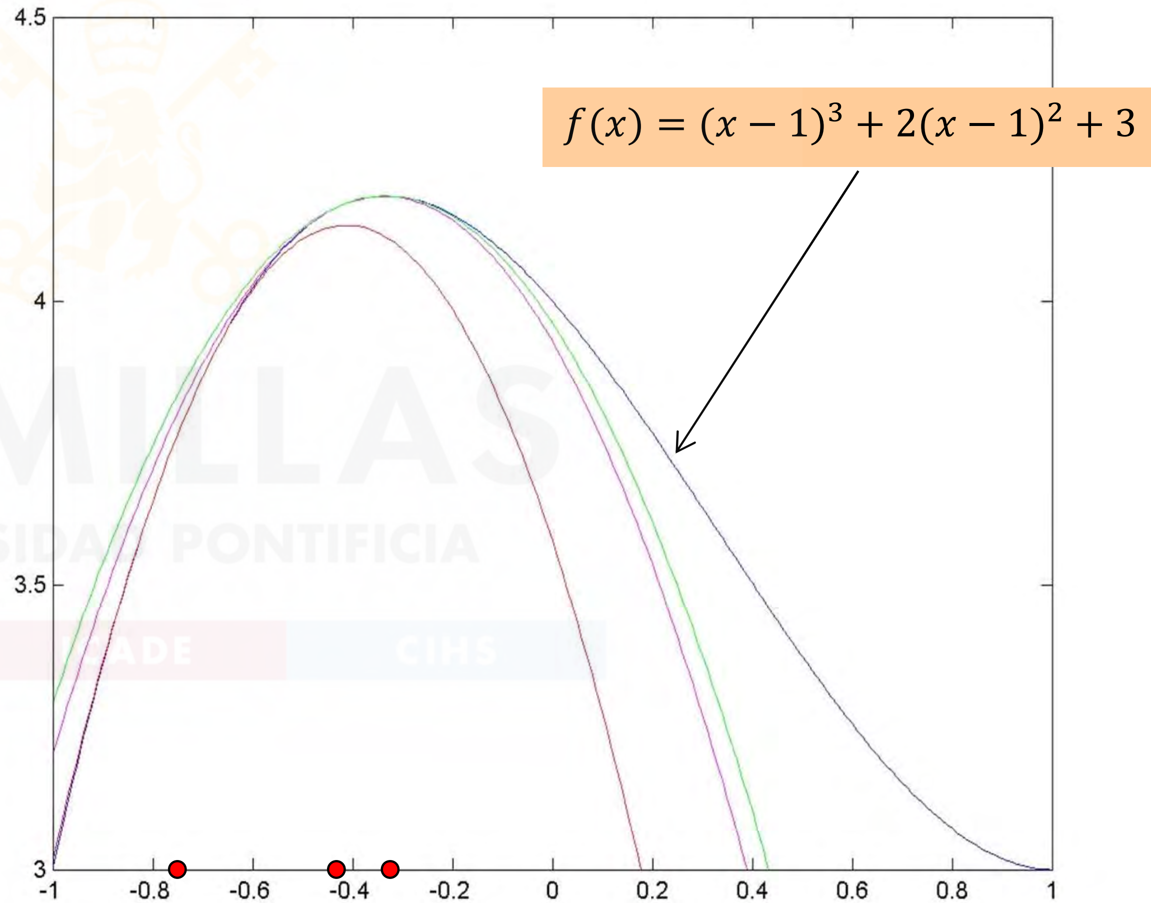
Sequence of points

$x_0 = -0.75$

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = -0.4135$$

$x_2 = -0.3376$

$x_3 = -0.3333$



General nonlinear optimization procedure

- Generate a sequence of points until convergence
  - Start from an initial point x_k
  - Obtain a search direction p_k
  - Calculate the step size α_k
  - Update the new point x_{k+1}

$$x_{k+1} = x_k + \alpha_k p_k$$

Conditions for each iteration

$$x_{k+1} = x_k + \alpha_k p_k$$

- Choose the new point such that the value of the objective function decreases $f(x_{k+1}) < f(x_k)$
- Conditions for the search direction p_k
  - The direction is descending $p^T \nabla f(x_k) < 0$
  - The descent is sufficient (non orthogonal vectors) $-\frac{p^T \nabla f(x_k)}{\|p\| \cdot \|\nabla f(x_k)\|} \geq \varepsilon > 0$
  - The search direction is related to the gradient $\|p\| \geq m \|\nabla f(x_k)\|$
- Conditions for the scalar α_k
  - The descent is sufficient (Armijo condition)

    $$f(x_{k+1}) \leq f(x_k) + \mu \alpha_k p_k^T \nabla f(x_k) \qquad 0 < \mu < 1$$

  - The descent is not too small

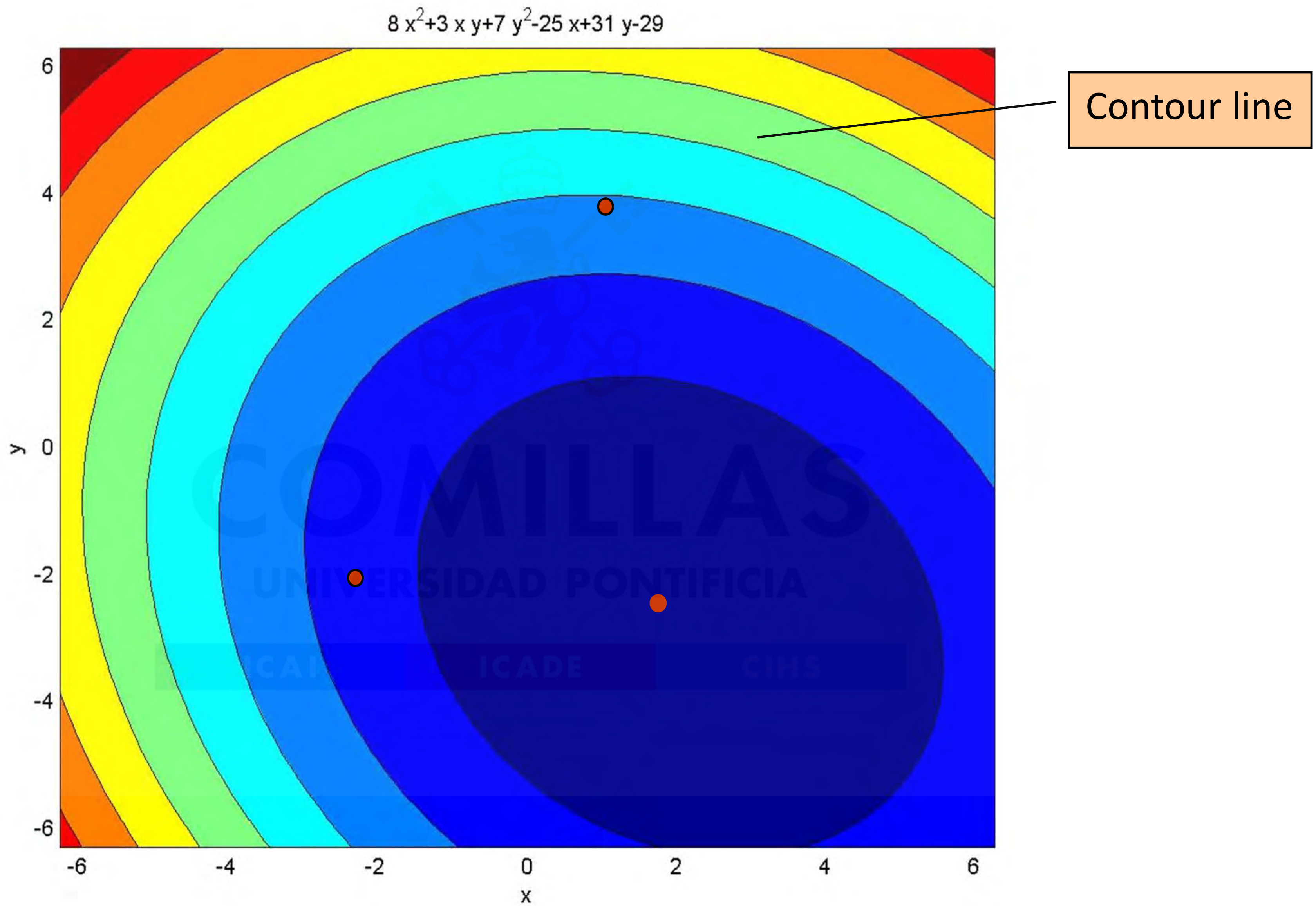
    For example, α_k is defined like a sequence 1, ½, ¼, 1/8, etc. The values of the sequence are used starting with 1. If for $\alpha = 1$ the previous condition is satisfied, we stop, otherwise the next value of the sequence is used.
  - Minimize the value of the function: one-dimensional search method (*line-search methods*)

    $$\min_{\alpha > 0} F(\alpha) \equiv f(x_k + \alpha p_k)$$

Rosenbrock method or method of cyclic coordinates

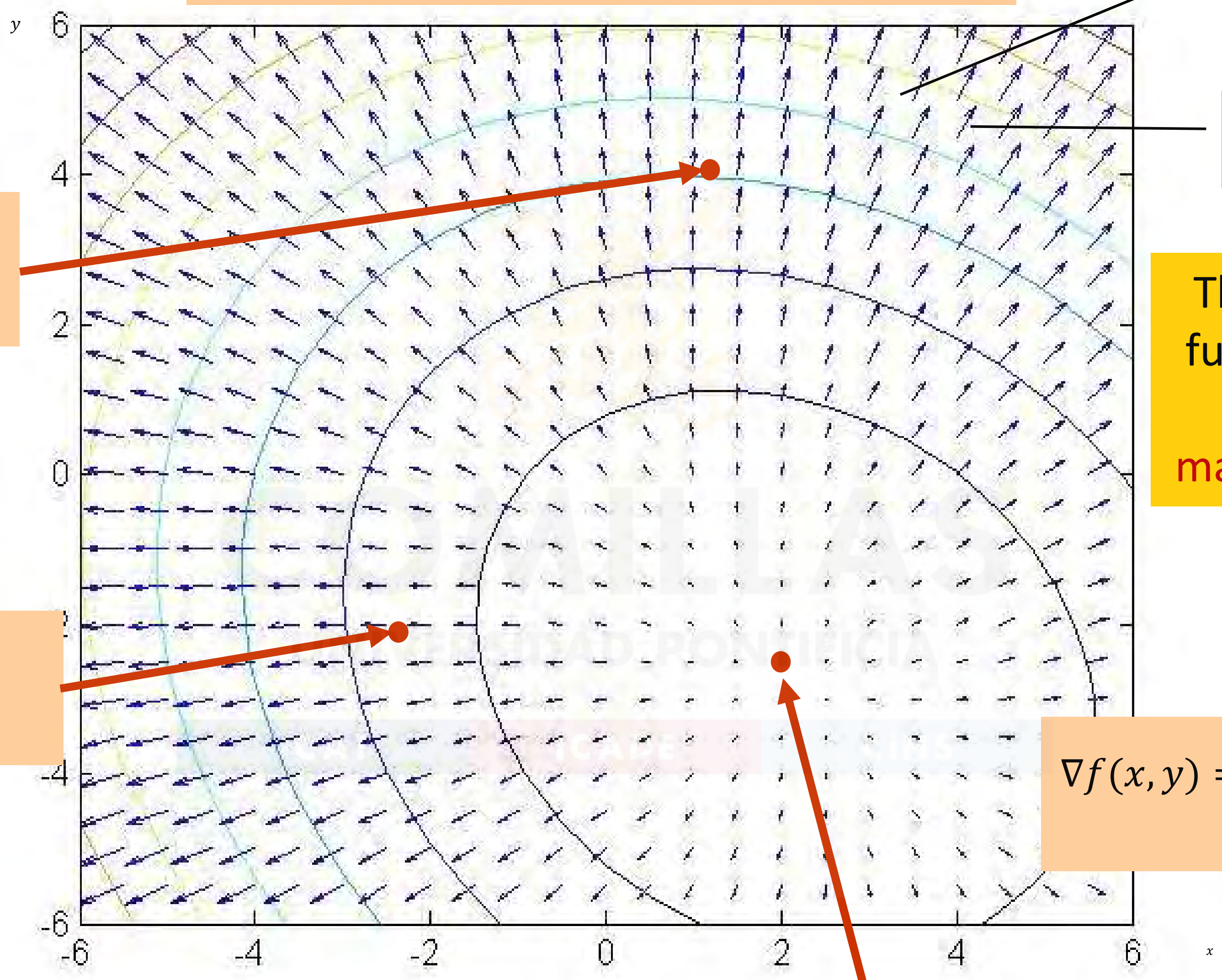
- This methods starts from a point x_1 and minimizes the function f in the direction $d_1 = (1,0,\ldots,0)$ (minimization of a one-dimensional function); reaching the point x_2 which minimizes the function in this direction, we minimize starting from this point in the direction $d_2 = (0,1,\ldots,0)$ in order to obtain the point x_3 and so on and so forth until we reach the point x_{n+1} where we minimize in the direction $d_1 = (1,0,\ldots,0)$ again.
- The process is repeated until we obtained the desired precision. This is not a very efficient method; it does not take great advantage of the descent directions, but this method is good in order to obtain a first intuition about search methods. Every one-dimensional search can be with or without derivatives.

Gradient (i)



Gradient (ii)

$f(x, y) = 8x^2 + 3xy + 7y^2 - 25x + 31y - 29$



Contour line

Gradient vector

$\nabla f(2,3) = \begin{pmatrix} 5 \\ 77.5 \end{pmatrix}$

$\nabla f(-2,-2) = \begin{pmatrix} -63 \\ -3 \end{pmatrix}$

$(x, y) = (2.060, -2.656)$

$\nabla f(x, y) = \begin{pmatrix} 16x + 3y - 25 \\ 3x + 14y + 31 \end{pmatrix}$

The gradient of a function gives the direction of maximum increase

Method of the steepest descent

- It does not require the use of second derivatives. Hence, it does not require heavy computation.
- Has a linear convergence rate (quite slow).
- In general, it should not be used.
- The function is approximated by a first order Taylor series
- The resulting search direction in order to minimize the function is the direction opposite to the gradient of the function.

$$p_k = -\nabla f(x_k)$$

Example 1

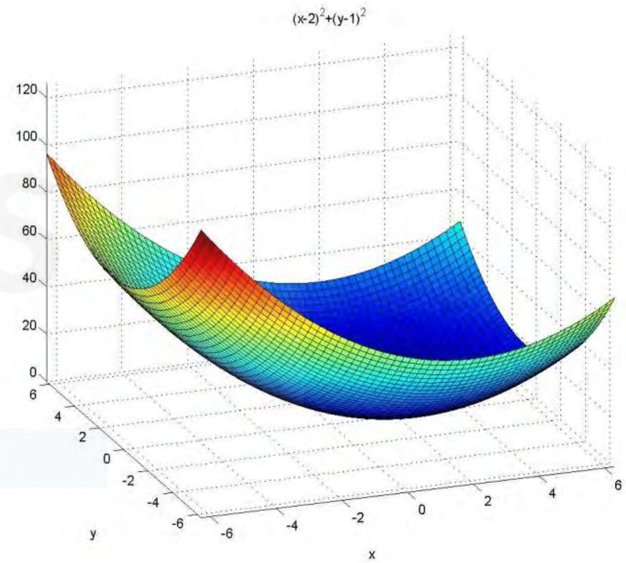
$f(x,y) = (x-2)^2 + (y-1)^2$

$p_k = -\nabla f(x_k, y_k) = -\begin{pmatrix} 2(x_k - 2) \\ 2(y_k - 1) \end{pmatrix}$

$\min_{\alpha>0} F(\alpha) = f(x_k + \alpha p_k)$

- Initial point x_k=(0,0)
- Search direction p_k=(4,2)

$$\min_{\alpha} F(\alpha) = (4\alpha - 2)^2 + (2\alpha - 1)^2$$
$$= 20\alpha^2 - 20\alpha + 5$$
$$F'(\alpha) = 0$$
$$40\alpha - 20 = 0$$
$$\alpha = 0.5$$



- Next point $x_{k+1} = x_k + \alpha_k p_k$=(2,1)
- Search direction p_k=(0,0)
- Since the gradient is 0, we have arrived at the optimum

Example 2 (i)

$$f(x,y) = 8x^2 + 3xy + 7y^2 - 25x + 31y - 29$$

$$p_k = -\nabla f(x_k, y_k) = -\begin{pmatrix} 16x_k + 3y_k - 25 \\ 3x_k + 14y_k + 31 \end{pmatrix}$$

$$\min_{\alpha} F(\alpha) = f(x_k + \alpha p_k)$$

- Initial point x_k=(-4,4)
- Search direction p_k=(77,-75)

$$\min_{\alpha} F(\alpha)$$

$$= \begin{matrix} 8(-4+77\alpha)^2 + 3(-4+77\alpha)(4-75\alpha) + \\ 7(4-75\alpha)^2 - 25(-4+77\alpha) + 31(4-75\alpha) - 29 = \\ 69482\alpha^2 - 11554\alpha + 387 \end{matrix}$$

$$F'(\alpha) = 0$$

$$138964\alpha - 11554 = 0$$

$$\alpha = 0.0831$$

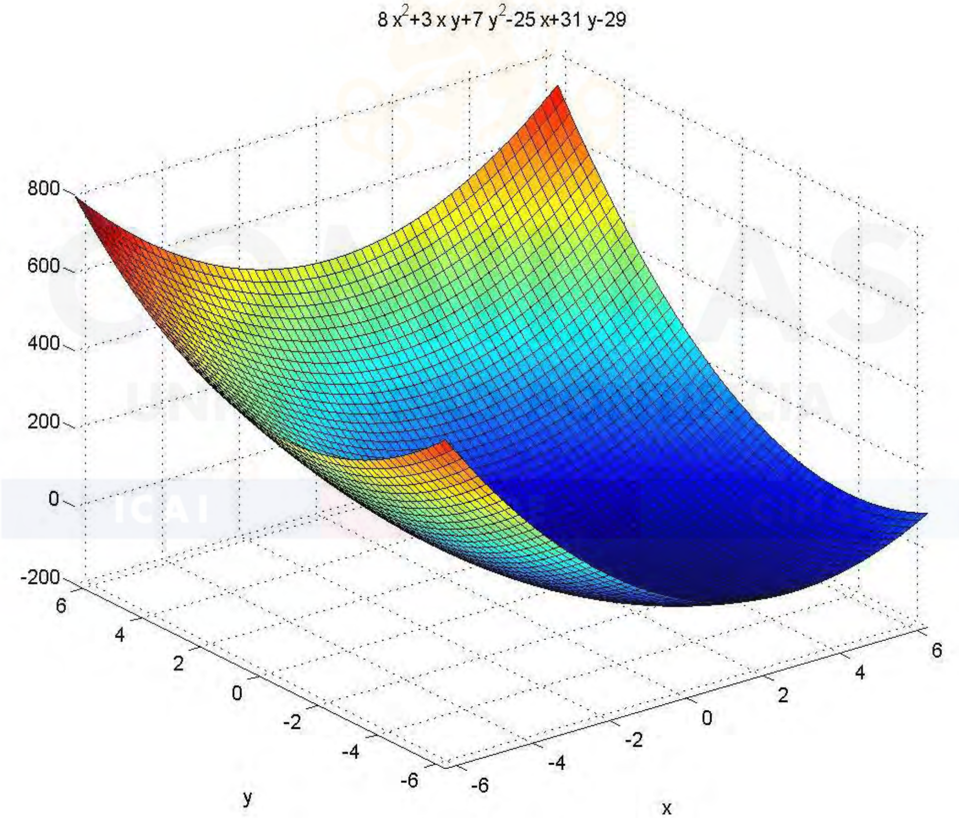
- Next point (-4+0.0831x77,4-0.0831x75)=(2.402,-2.236)

Example 2 (ii)

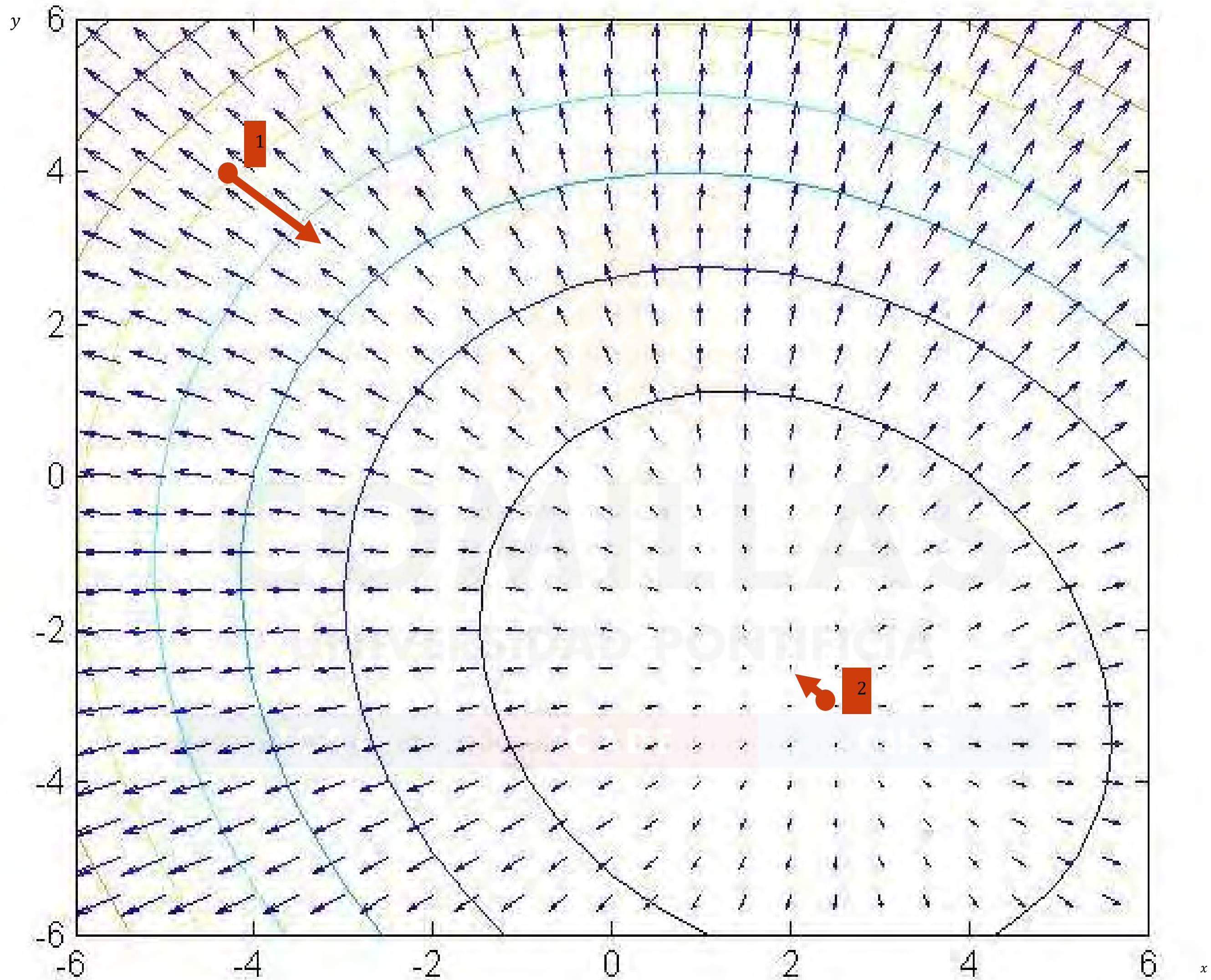
- Search direction p_k=(-6.726,-6.905)

$$\min_{\alpha} F(\alpha) = \begin{array}{c} 8(2.4-6.7\alpha)^2 + 3(2.4-6.7\alpha)(-2.2-6.9\alpha) + \\ 7(-2.2-6.9\alpha)^2 - 25(2.4-6.7\alpha) + 31(-2.2-6.9\alpha) - 29 \end{array}$$

- Etc.



Example 2 (iii)



Example 3 (i)

$$\min_x f(x) = \frac{1}{2} x^T Q x - b^T x$$

$$Q = \begin{pmatrix} 1 & \cdot & \cdot \\ \cdot & 5 & \cdot \\ \cdot & \cdot & 25 \end{pmatrix}$$

$$b = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

- For quadratic functions, the optimum can be obtained by $\nabla f(x) = Qx - b = 0$ and, therefore, $x^* = Q^{-1} b = \begin{pmatrix} -1 \\ -\frac{1}{5} \\ -\frac{1}{25} \end{pmatrix}$
- The direction of steepest descent is $p_k = -\nabla f(x_k) = -(Qx_k - b)$
- If we use an exact line-search, the resulting value is

$$\alpha_k = -\frac{\nabla f(x_k)^T p_k}{p_k^T Q p_k}$$

Example 3 (ii)

- Initial point $x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $f(x_0) = 0$ $\nabla f(x_0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ $\alpha_0 = 3/31 = 0.097$

- As convergence measure we use the 2-norm of the gradient

$$\|\nabla f(x_0)\| = \sqrt{1^2 + 1^2 + 1^2} = 1.73$$

- New point

$$x_1 = x_0 + \alpha_0 p_0 = \begin{pmatrix} -0.097 \\ -0.097 \\ -0.097 \end{pmatrix} \quad f(x_1) = -0.145 \quad \nabla f(x_1) = \begin{pmatrix} 0.903 \\ 0.516 \\ -1.419 \end{pmatrix} \quad \alpha_1 = 0.059$$

- 2-norm of the gradient $\|\nabla f(x_1)\| = 1.760$

Example 3 (iii)

- New point

$$x_2 = \begin{pmatrix} -0.150 \\ -0.127 \\ -0.013 \end{pmatrix}$$

$$f(x_2) = -0.237$$

$$\nabla f(x_2) = \begin{pmatrix} 0.850 \\ 0.364 \\ 0.673 \end{pmatrix}$$

- 2-norm of the gradient $\|\nabla f(x_2)\| = 1.144$
- Etc.

- The process continues until the 2-norm of the gradient is sufficiently small (smaller than a certain tolerance, 10^{-8} for example). In this example we need 216 iterations until we reach this tolerance.

Convergence rate of the method of steepest descent for quadratic functions

- The convergence rate of the steepest descent method for a quadratic function with one-dimensional exact search is linear. The relation of improvement between two consecutive iterations is upper bounded by:

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \le \left(\frac{\text{cond}(Q) - 1}{\text{cond}(Q) + 1}\right)^2$$

where the condition number of a matrix A is defined as $\text{cond}(A) \equiv \|A\| \cdot \|A^{-1}\|$ and $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$, where $\lambda_{\max}(A^T A)$ is the maximum eigenvalue of matrix $A^T A$.

- If A is a symmetric, positive definite matrix then $\text{cond}(A) = \lambda_1/\lambda_n$ where λ_1 and λ_n are the largest and the smallest eigenvalue of the matrix. For the previous example: $\text{cond}(Q) = 25$
- If the condition number is large, this indicates a slow convergence rate. For $\text{cond}(Q) = 100$ this method would improve its solution by at most 4 % in each iteration.

Condition number

- The condition number is a measure for the sensitivity of the solution of a system of linear equations to errors in the data, stability of the solution
- Condition numbers below 10^6 are good enough. Numerical problems arise for condition numbers above 10^{10}
- Can be shown in the process log
  - Quality parameter for CPLEX and Kappa for GUROBI

Convergence rate of the method of steepest descent for general nonlinear functions

- For general nonlinear functions the convergence of this method is also linear but with the following upper bound

$$\left(\frac{\mathrm{cond}(Q)-1}{\mathrm{cond}(Q)+1}\right)^2$$

where $Q = \nabla^2 f(x^*)$ is the Hessian of the function in the solution.

Newton method for solving a system of nonlinear equations (i)

- Solves a system of nonlinear equations iteratively

$$
\begin{aligned}
f_1(x_1, \dots, x_n) &= 0 \\
f_2(x_1, \dots, x_n) &= 0 \\
&\vdots \\
f_n(x_1, \dots, x_n) &= 0
\end{aligned}
$$

- Approximates the nonlinear function by a linear function in each point (iteration), using a first order Taylor expansion

$$
\begin{aligned}
& f(x_k + p_k) \approx f(x_k) + \nabla f(x_k)^T p_k \\
& f(x^*) \approx f(x_k) + \nabla f(x_k)^T p_k = 0 \\
& p_k = -\nabla f(x_k)^{-T} f(x_k) \\
& x_{k+1} = x_k + p_k \\
& = x_k - \nabla f(x_k)^{-T} f(x_k)
\end{aligned}
$$

- $\nabla f(x)^T = (\nabla f_1(x) \quad \nabla f_2(x) \quad \dots \quad \nabla f_n(x))^T$ Jacobian of the function

Newton method for a system of nonlinear equations (ii)

- Has quadratic convergence if the point is close to the solution
- The Jacobian of the function has to be non singular in each point

Newton method for a system of nonlinear equations (iii)

$$f_1(x_1, x_2) = 3x_1x_2 + 7x_1 + 2x_2 - 3 = 0$$
$$f_2(x_1, x_2) = 5x_1x_2 - 9x_1 - 4x_2 + 6 = 0$$

$$\nabla f(x_1, x_2) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 3x_2 + 7 & 5x_2 - 9 \\ 3x_1 + 2 & 5x_1 - 4 \end{pmatrix}$$

$$x_{k+1} = x_k - \begin{pmatrix} 3x_2 + 7 & 5x_2 - 9 \\ 3x_1 + 2 & 5x_1 - 4 \end{pmatrix}^{-T} \begin{pmatrix} 3x_1x_2 + 7x_1 + 2x_2 - 3 \\ 5x_1x_2 - 9x_1 - 4x_2 + 6 \end{pmatrix}$$

- Consider this as the initial point $x_0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$$x_1 = x_0 - \begin{pmatrix} 3x_2 + 7 & 5x_2 - 9 \\ 3x_1 + 2 & 5x_1 - 4 \end{pmatrix}^{-T} \begin{pmatrix} 3x_1x_2 + 7x_1 + 2x_2 - 3 \\ 5x_1x_2 - 9x_1 - 4x_2 + 6 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 13 & 1 \\ 5 & 1 \end{pmatrix}^{-T} \begin{pmatrix} 14 \\ -1 \end{pmatrix} = \begin{pmatrix} -1.375 \\ 5.375 \end{pmatrix}$$

- After 8 iterations the point takes the value of $x_8 = \begin{pmatrix} 0 \\ 1.5 \end{pmatrix}$ and the functions take value 0.

Newton method for optimization

- The Newton method for a system of nonlinear equations is applied to the first order necessary optimality conditions.

$$\nabla f(x) = 0$$

- The Jacobian of this function $\nabla f(x)$ is the Hessian $\nabla^2 f(x)$
- Iteration $x_{k+1} = x_k + p_k = x_k - \nabla^2 f(x_k)^{-T} \nabla f(x_k)$
- Where p_k (the Newton direction) is obtained by solving a system of linear equations (Newton system) instead of calculating the inverse of the Hessian.

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

Example 1 (i)

$$f(x,y) = (x-2)^2 + (y-1)^2$$

$$\nabla f(x_k, y_k) = \begin{pmatrix} 2(x_k-2) \\ 2(y_k-1) \end{pmatrix}$$

$$\nabla^2 f(x_k, y_k) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

- Initial point x_k=(0,0)
- Search direction p_k=(2,1)

$$p_k = -\begin{pmatrix} 2 & \cdot \\ \cdot & 2 \end{pmatrix}^{-T} \begin{pmatrix} -4 \\ -2 \end{pmatrix} = -\begin{pmatrix} 0.5 & \cdot \\ \cdot & 0.5 \end{pmatrix} \begin{pmatrix} -4 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

- Calculate α_k

$$x_{k+1} = x_k + \alpha_k p_k$$

$$\min_{\alpha} F(\alpha) = (2\alpha - 2)^2 + (\alpha - 1)^2$$
$$= 5\alpha^2 - 10\alpha + 5$$
$$F'(\alpha) = 0$$
$$10\alpha - 10 = 0$$
$$\alpha = 1$$

Example 1 (ii)

- Next point $x_{k+1} = x_k + \alpha_k p_k$ = (2,1)
- Search direction p_k = (0,0)
- We have arrived at the optimum since the gradient is 0

Quasi-Newton method

- Decrease the computational cost associated with calculating and storing the Hessian and with solving the system of linear equations.
- Based on approximating the Hessian of the function $\nabla^2 f(x_k)$ in each point by positive definite matrix B_k which is easier to calculate. The different quasi-Newton methods vary in their choice of B_k and in the way of updating this matrix.
- Advantages:
  - Does not have to calculate second derivatives (Hessian), instead uses only first derivatives in the approximation of B_k
  - The search direction can be calculated with small computational cost
- Disadvantages:
  - The convergence is not quadratic anymore, iterations are less costly
  - Requires storing a matrix, hence these methods are not adequate for large problems

Calculating the matrix B_k

- Condition of the secant $\nabla^2 f(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1})$
- Approximation $B_k(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1})$
- For a quadratic function, the Hessian Q satisfies this condition, hence the approximation is exact.
- Defining $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ then $B_{k+1} s_k = y_k$
- In quasi-Newton methods the matrix B_k is updated in each iteration

$$B_{k+1} = B_k + \text{update}$$

- The initialization of B_k usually is the identity matrix $B_0 = I$.

Quasi-Newton procedure

1. Specify an initial solution x_0 and an initial approximation of the Hessian B_0
2. Iterate k=0,1,… until finding the optimal solution
   - Calculate the search direction $B_k p_k = -\nabla f(x_k)$
   - Carry out a line-search to determine $x_{k+1} = x_k + \alpha_k p_k$
   - Calculate $s_k = x_{k+1} - x_k$ $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
   - Update the approximation of the Hessian

$$B_{k+1} = B_k + \text{update}$$

Updating the matrix B_k

- Update BFGS (Broyden-Fletcher-Goldfarb-Shanno)

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

- Update DFP (Davidon-Fletcher-Powell)

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + (s_k^T B_k s_k) u_k u_k^T$$

$$u_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}$$

- In order to guarantee that B_{k+1} is still positive definite, both updates have to fulfill that $y_k^T s_k > 0$ a condition which has to be guaranteed controlling the one-dimensional line-search.
- In each iteration the matrix B_k approximates the Hessian better

NLP PROBLEMS
TYPE OF NLP PROBLEMS
CLASSIFICATION OF UNCONSTRAINED METHODS
OPTIMALITY CONDONDITIONS FOR NONLINEAR UNCONSTRAINED OPTIMIZATION
OPTIMALITY CONDITIONS FOR NLP
METHODS FOR UNCONSTRAINED OPTIMIZATION (master)
NONLINEAR PROGRAMMING METHODS (master)

7

NONLINEAR PROGRAMMING METHODS (master)

Nonlinear programming

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) \leq 0 \quad i = 1,\ldots,m$$
$$h_j(x) = 0 \quad j = 1,\ldots,l$$

$f, g_i, h_j: \mathbb{R}^n \to \mathbb{R}$ $\quad x \in \mathbb{R}^n$

where *f*, *g* and *h* are differentiable functions with first and second continuous derivatives.

- If the feasible region is convex and the objective function is convex in all of the feasible region, the local optimum is also a global optimum

Nonlinear programming methods (NLP)

1. Penalty methods: minimize a function related to the Lagrangian that has the same minimum
    - Penalty methods and barrier methods
    - Augmented Lagrangian method
    - Sequential quadratic programming method
2. Feasible point methods: maintain feasibility starting from a feasible point and moving to feasible directions
    - Generalization of the simplex method of LP. Solve a sequence of subproblems with a set of binding constraints which change in each iteration.
    - Disadvantages: the selection of the set of binding constraints and the difficulty of satisfying the constraints
        - Reduced gradient method

Penalty methods

- Solve a NLP problem by solving a sequence of unconstrained optimization problems. At the limit, the solution of both problems is the same.
- In the objective function we include penalties that measure the violation of the constraints and also some parameters that determine the importance of each constraint.
  - Penalty methods (Exterior penalty method)
    - Penalize the violation of a constraint. Moves along feasible points.
    - Works better for equality constraints.
  - Barrier methods (Interior penalty method)
    - Avoids reaching the limits (outline) of a constraint. Strictly feasible points.
    - Do not work for equality constraints.

Penalty method (exterior)

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) = 0 \quad i = 1, \dots, m$$

- Let us define the penalty function

$$\psi(x) = \frac{1}{2}\sum_{i=1}^{m} g_i(x)^2 = \frac{1}{2} g(x)^T g(x)$$

- The original problem is transformed into

$$\min_x \pi(x, \rho_k) = f(x) + \rho_k \psi(x)$$

where ρ_k is the penalty parameter, a positive scalar which grows monotonically to ∞

- For advanced iterations, the optimum moves to the feasible region

Penalty method (exterior)

- Optimality condition of the penalty function

$$\nabla f(x(\rho)) + \rho \sum_{i=1}^{m} \nabla g_i(x(\rho)) g_i(x(\rho)) = 0$$

- Let us define $\lambda_i = \lambda_i(\rho) = \rho g_i(x(\rho))$. Estimation of the Lagrange multipliers.
- Hence, the optimality conditions of the original function are

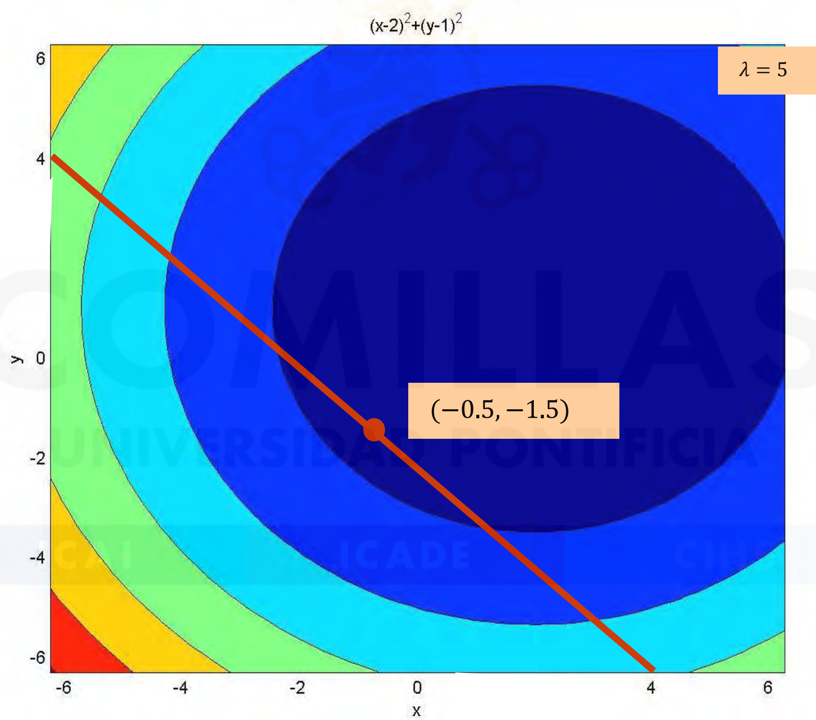
$$\nabla f(x(\rho)) + \sum_{i=1}^{m} \lambda_i(\rho) \nabla g_i(x(\rho)) = 0$$
$$\lambda_i(\rho) = \rho g_i(x(\rho)) \quad i = 1, \dots, m$$
$$\lambda_i(\rho) \geq 0 \quad i = 1, \dots, m$$

Penalty method (exterior)

- The Hessian of the penalized problem can give a large condition number $\nabla_x^2 \pi(x(\rho), \rho)$

Example 1 (i)

$$\min (x-2)^2 + (y-1)^2$$
$$x + y = -2$$



Example 1 (ii)

$$\min (x-2)^2 + (y-1)^2 + \frac{1}{2}\rho(x+y+2)^2$$

- First order optimality condition $\nabla_x \pi(x) = 0$. System of 2 equations

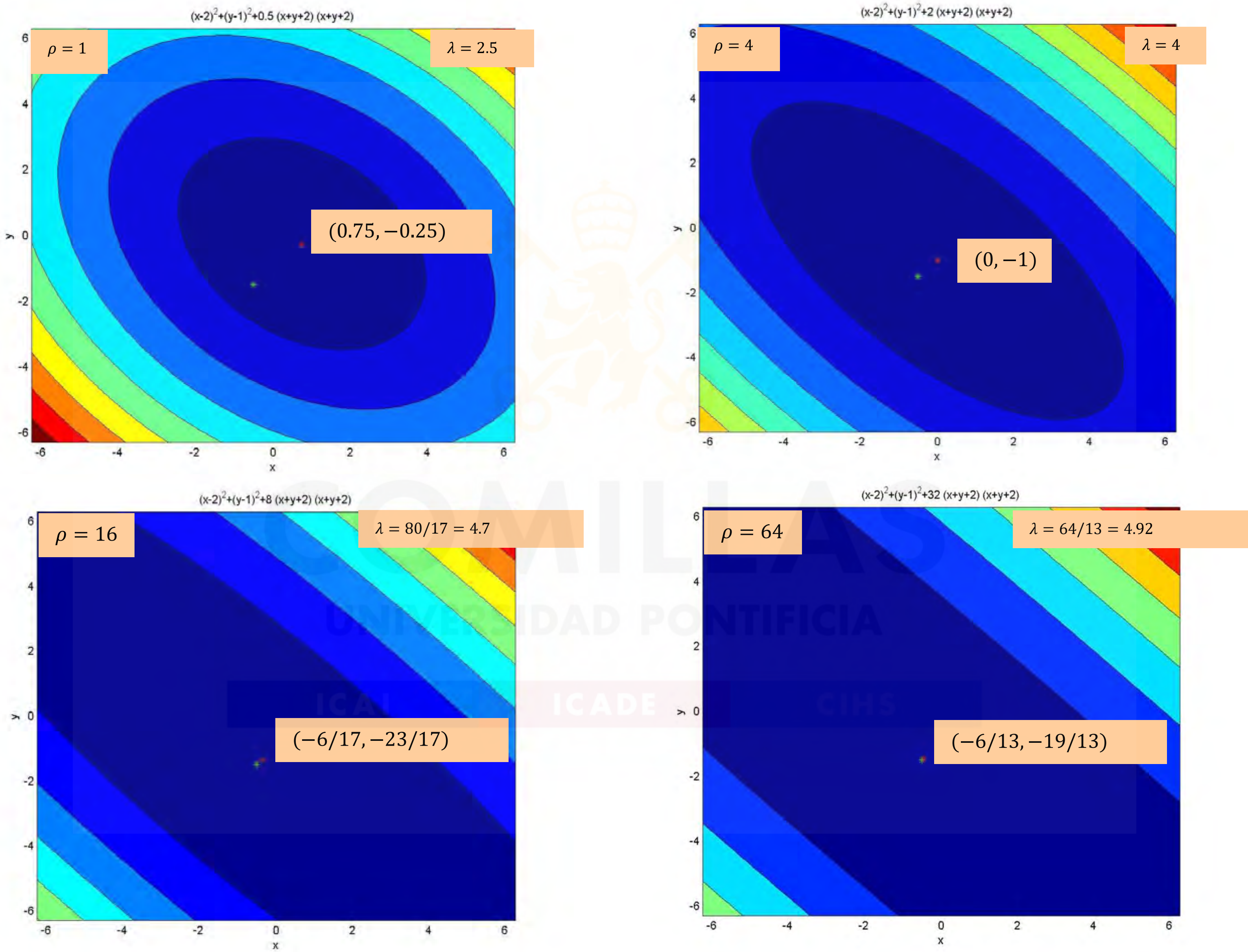
$$2(x-2) + \rho(x+y+2) = 0$$
$$2(y-1) + \rho(x+y+2) = 0$$

$$x = \frac{4-\rho}{2+2\rho} \qquad y = \frac{2-3\rho}{2+2\rho} \qquad \lambda = \frac{10\rho}{2+2\rho}$$

- For ρ=1 $(x^*, y^*) = (0.75, -0.25)$ $\lambda = 2.5$
- For ρ=2 $(x^*, y^*) = (1/3, -2/3)$
- For ρ=4 $(x^*, y^*) = (0, -1)$ $\lambda = 4$
- For ρ=8 $(x^*, y^*) = (-2/9, -11/9)$
- For ρ=16 $(x^*, y^*) = (-6/17, -23/17)$ $\lambda = 80/17 = 4.7$
- For ρ=32 $(x^*, y^*) = (-14/33, -47/33)$
- For ρ=64 $(x^*, y^*) = (-6/13, -19/13)$ $\lambda = 64/13 = 4.92$
- For ρ=∞ $(x^*, y^*) = (-0.5, -1.5)$ $\lambda = 5$

Each of these problems is an unconstrained optimization

Example 1 (iii)



Example 1 (iv)

- The Hessian of the penalized problem is $\nabla_x^2 \pi(x(\rho),\rho) = \begin{pmatrix} 2+\rho & \rho \\ \rho & 2+\rho \end{pmatrix}$

- For ρ=1 $\nabla_x^2 \pi(x(\rho),\rho) = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$ $cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 4/2 = 2$

- For ρ=4 $\nabla_x^2 \pi(x(\rho),\rho) = \begin{pmatrix} 6 & 4 \\ 4 & 6 \end{pmatrix}$ $cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 10/2 = 5$

- For ρ=16 $\nabla_x^2 \pi(x(\rho),\rho) = \begin{pmatrix} 18 & 16 \\ 16 & 18 \end{pmatrix}$ $cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 34/2 = 17$

- For ρ=64 $\nabla_x^2 \pi(x(\rho),\rho) = \begin{pmatrix} 66 & 64 \\ 64 & 66 \end{pmatrix}$ $cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 130/2 = 65$

Barrier method (interior penalization)

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) \geq 0 \quad i = 1, \dots, m$$

- Let us define the barrier function

$$\varphi(x) = -\sum_{i=1}^{m} \log(g_i(x))$$

or rather

$$\varphi(x) = \sum_{i=1}^{m} \frac{1}{g_i(x)}$$

  - It is a continuous function in the interior of the feasible region which goes to ∞ when getting close to the boundary

- The original problem is transformed into

$$\min_x \pi(x, \mu_k) = f(x) + \mu_k \phi(x)$$

where μ_k, the barrier parameter, is a positive scalar which decreases monotonically to 0

- When decreasing the parameter, the points get closer to the boundary of the feasible region

Penalty method (exterior)

- Optimality condition of the barrier function

$$\nabla f(x) - \mu \sum_{i=1}^{m} \frac{\nabla g_i(x)}{g_i(x)} = 0$$

- Let us define $\lambda_i = \lambda_i(\mu) = \dfrac{-\mu}{g_i(x)}$ $\quad \lambda_i(\mu) g_i(x) = -\mu$.
- Estimation of the Lagrange multipliers.
- Hence, the optimality conditions of the original function are

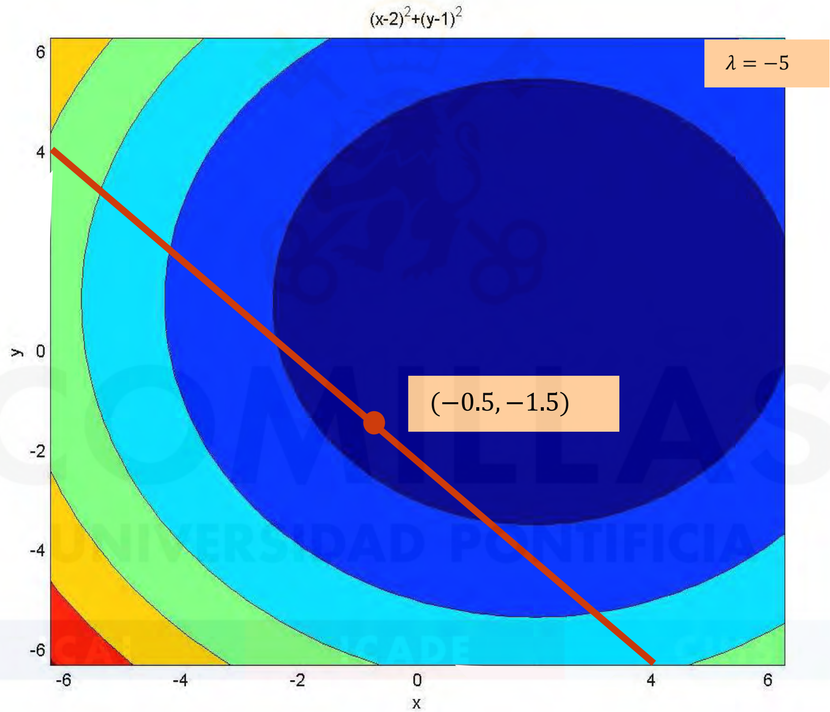
$$\nabla f(x(\mu)) + \sum_{i=1}^{m} \lambda_i(\mu) \nabla g_i(x(\mu)) = 0$$
$$\lambda_i(\mu) g_i(x(\mu)) = -\mu \quad i = 1, \dots, m$$
$$\lambda_i(\mu) \leq 0 \quad i = 1, \dots, m$$

Penalty method (exterior)

- The Hessian of the penalized problem can give a large condition number $\nabla_x^2 \pi(x(\mu), \mu)$

Example 1 (i)

$$\min (x-2)^2 + (y-1)^2$$
$$-x - y \geq 2$$



Example 1 (ii)

$$\min (x-2)^2 + (y-1)^2 - \mu \log(-x-y-2)$$

- First order optimality conditions $\nabla_x \pi(x) = 0$. System of 2 equations

$$2(x-2) + \mu/(-x-y-2) = 0$$
$$2(y-1) + \mu/(-x-y-2) = 0$$

$$x = \frac{3}{4} \pm \frac{1}{8}\sqrt{100+16\mu}$$

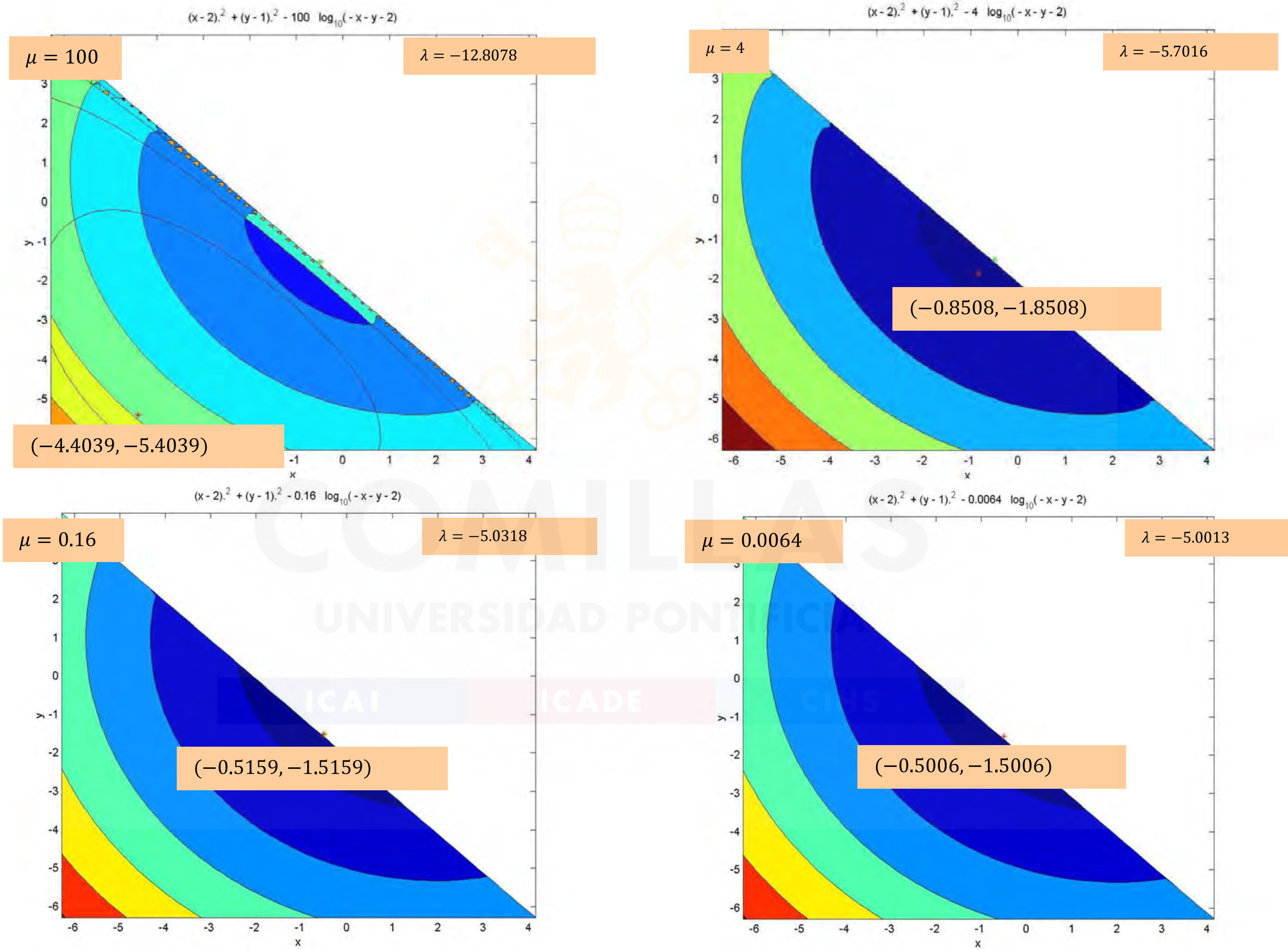
$$y = -\frac{1}{4} \pm \frac{1}{8}\sqrt{100+16\mu}$$

$$\lambda = \frac{\mu}{x+y+2}$$

- For μ=100 $(x^*, y^*) = (-4.4039, -5.4039)$ $\lambda = -12.8078$
- For μ=20 $(x^*, y^*) = (-1.8117, -2.8117)$
- For μ=4 $(x^*, y^*) = (-0.8508, -1.8508)$ $\lambda = -5.7016$
- For μ=0.8 $(x^*, y^*) = (-0.5776, -1.5776)$
- For μ=0.16 $(x^*, y^*) = (-0.5159, -1.5159)$ $\lambda = -5.0318$
- For μ=0.032 $(x^*, y^*) = (-0.5032, -1.5032)$
- For μ=0.0064 $(x^*, y^*) = (-0.5006, -1.5006)$ $\lambda = -5.0013$
- For μ=0 $(x^*, y^*) = (-0.5, -1.5)$ $\lambda = -5$

Each of these problems is an unconstrained optimization

Example 1 (iii)



Example 1 (iv)

- The Hessian of the barrier problem is

$$\nabla_x^2 \pi(x(\mu),\mu) = \begin{pmatrix} 2+\mu/(x+y+2)^2 & \mu/(x+y+2)^2 \\ \mu/(x+y+2)^2 & 2+\mu/(x+y+2)^2 \end{pmatrix}$$

- For μ=100

$$\nabla_x^2 \pi(x(\mu),\mu) = \begin{pmatrix} 3.64 & 1.64 \\ 1.64 & 3.64 \end{pmatrix}$$

$$cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 5.28/2 = 2.64$$

- For μ=4

$$\nabla_x^2 \pi(x(\mu),\mu) = \begin{pmatrix} 10.127 & 8.127 \\ 8.127 & 10.127 \end{pmatrix}$$

$$cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 18.25/2 = 9.125$$

- For μ=0.16

$$\nabla_x^2 \pi(x(\mu),\mu) = \begin{pmatrix} 160.24 & 158.24 \\ 158.24 & 160.24 \end{pmatrix}$$

$$cond(\nabla^2 \pi) = \lambda_1/\lambda_2 = 318.48/2 = 159.24$$

Augmented Lagrangian method (i)

- The ill conditioning of the previous methods can be improved by including the multipliers in the penalty function.
- The algorithms therefore have to update both, the variables and the multipliers.
- The convergence rate is faster than in the previous methods.

Augmented Lagrangian method (ii)

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) = 0 \quad i = 1,\dots,m$$

- The optimum of the Lagrangian coincides with the optimum of the previous problem if it is a feasible point

$$\min_x L(x) = f(x) + \sum_{i=1}^{m} \lambda g_i(x)$$
$$g_i(x) = 0 \quad i = 1,\dots,m$$

- We solve this problem using the exterior penalty method

$$\min_x A(x) = f(x) + \sum_{i=1}^{m} \lambda_i^k g_i(x) + \frac{1}{2}\rho_k \sum_{i=1}^{m} [g_i(x)]^2$$

- In general form

$$\min A(x,\lambda,\rho) = f(x) + \lambda^T g(x) + \frac{1}{2}\rho g(x)^T g(x)$$

Augmented Lagrangian method (iii)

- Choose values for x_0 and λ_0 ρ_0
- Check optimality. If verified, then the algorithm stops.

$$\nabla L(x_k, \lambda_k) = 0$$

- Solve the nonlinear unconstrained problem and calculate x_{k+1}

$$\min_x A(x, \lambda_k, \rho_k) = f(x) + \lambda_k^T g(x) + \frac{1}{2}\rho_k g(x)^T g(x)$$

- Update λ_{k+1} and ρ_{k+1}

$$\lambda_{k+1} = \lambda_k + \rho_k g(x_{k+1})$$

ρ_{k+1} has to be greater than ρ_k

Sequential quadratic programming (i)

- Consider the problem

$$\min_x f(x)$$
$$g_i(x) = 0 \quad i = 1, \ldots, m$$

- Let us formulate the Lagrangian

$$\min_x L(x) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$

- First order optimality conditions

$$\nabla L(x^*, \lambda^*) = 0 \quad \Rightarrow \quad \begin{cases} \nabla_x L(x, \lambda) = \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0 \\ \nabla_\lambda L(x, \lambda) = g_i(x) = 0 \end{cases}$$

- The Newton method is formulated for this system of equations

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} p_k \\ v_k \end{pmatrix}$$

where p_k and v_k are determined by solving the following system of linear equations.

Sequential quadratic programming (ii)

- System of linear equations

$$\nabla^2 L(x_k, \lambda_k) \binom{p_k}{v_k} = -\nabla L(x_k, \lambda_k)$$

$$\begin{pmatrix} \nabla^2_{xx} L(x_k, \lambda_k) & \nabla g(x_k) \\ \nabla g(x_k)^T & 0 \end{pmatrix} \binom{p_k}{v_k} = \binom{-\nabla_x L(x_k, \lambda_k)}{g(x_k)}$$

- They represent the first order optimality conditions of this quadratic optimization problem with linear constraints

$$\min_p \frac{1}{2} p^T [\nabla^2_{xx} L(x_k, \lambda_k)] p + p^T [\nabla_x L(x_k, \lambda_k)]$$
$$[\nabla g(x_k)]^T p + g(x_k) = 0 \qquad : v_k$$

- Instead of solving a system of linear equations, this quadratic optimization problem with linear constraints is solved

Sequential quadratic programming(iii)

1. The following quadratic problem is solved

$$\min_p \frac{1}{2} p^T [\nabla^2_{xx} L(x_k, \lambda_k)] p + p^T [\nabla_x L(x_k, \lambda_k)]$$
$$[\nabla g(x_k)]^T p + g(x_k) = 0 \qquad : v_k$$

2. We obtain (p_k, v_k)
3. The values of the variables and the multipliers are updated

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} p_k \\ v_k \end{pmatrix}$$

Andrés Ramos

https://www.iit.comillas.edu/aramos/

Andres.Ramos@comillas.edu

Pedro Sánchez

Pedro.Sanchez@comillas.edu

Sonja Wogrin

https://www.iit.comillas.edu/swogrin/

Sonja.Wogrin@comillas.edu