CNA

Optimization.

V. Martín
FIM - UPM
Nov. 2011
v 0.0

Outline.

- Classical optimization techniques:
  - Introduction: Examples, Mathematical preliminaries.
  - Steepest descent methods.
  - Generating set search methods.
- Physically based algorithms::
  - Introduction: Simulated Annealing.
  - Monte Carlo Methods.

Outline

- Biologically based algorithms:
  - Evolutionary algorithms (more in the second part)
  - Immune networks.
- "Ecologically" based algorithms:
  - Ant foraging.
  - Flock algorithms.

Introduction to classical algorithms

- Example: stock management.
  - Calculate the investment in each of four options to get an expected profit above a 10%

option	Annual return rates.						
year	1	2	3	4	5	6	Average
Banks	18.24	12.12	15.23	5.26	2.62	10.42	10.64
Technology	12.24	19.16	35.07	23.46	-10.62	-7.43	11.98
Real state	8.23	8.96	8.35	9.16	8.05	7.29	8.34
Bonds	8.12	8.26	8.34	9.01	9.11	8.95	8.63

- Optimize an objective function with constraints.
  - Objective Function: Minimize the risk of losses.
    - A risk measurement is the fluctuation from the average value: variance. The variance of investment j is defined as:

$$v_{jj}=\frac{1}{n}\sum_{k=1}^{n}(r_{jk}-\mu_j)^2$$

  - r_{jk} = Return rate of investment j in year k.
  - μ_j = Average of investment j.

    - Example: Banks, from last table:

$$v_{11}=\frac{1}{6}[(18.24-10.64)^2+\cdots+(10.42-10.64)^2]=29.05$$

- The risk among different investment types is measured with the covariance matrix:

$$v_{ji} = \frac{1}{n} \sum_{k=1}^{n} (r_{jk} - \mu_j)(r_{ik} - \mu_i)$$

- Example: Banks vs. Technology stocks:

$$v_{12} = \frac{1}{6} [(18.24 - 10.64)(12.24 - 11.98) + \cdots + (10.42 - 10.64)(-7.43 - 11.98)] = 40.39$$

- The covariance matrix:

$$V = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix} = \begin{pmatrix} 29.055 & 40.39 & -0.28 & -1.95 \\ 40.39 & 267.34 & 6.83 & -3.69 \\ -0.28 & 6.83 & 0.37 & -0.06 \\ -1.95 & -3.69 & 0.05 & 0.15 \end{pmatrix}$$

- The covariance matrix allows to write the objective function (investment risk) as:

$$Risk = \vec{x}^T V \vec{x}$$

- Where $\vec{x} = (x_{1,} x_{2,} x_{3,} x_4)$ represents the fraction of investment in each of the stocks.
- Wich produces the objective function (to be minimized):

$$Risk = 29.05 x_1^2 + 80.78 x_2 x_1 - 0.57 x_3 x_1 - 3.90 x_4 x_1 + \\ + 267.34 x_2^2 + 0.37 x_3^2 + 0.15 x_4^2 + 13.66 x_2 x_3 + \\ - 7.39 x_2 x_4 - 0.11 x_3 x_4$$

– And the constraints are:

- Total investment is 1.

$$x_1 + x_2 + x_3 + x_4 = 1$$

- The return rate must be bigger than 10%

$$10.64\,x_1 + 11.98\,x_2 + 8.34\,x_3 + 8.63\,x_4 \geq 10$$

- All investments are positive.

$$x_i \geq 0 \quad i = 1, \cdots, 4$$

- The standard form for many optimization problems would be:
  - Find the values of $\vec{x}=(x_{1,} x_{2,} \cdots, x_n)^T$ to minimize the objective function $f(\vec{x})$ subject to constraints:

$g_i(\vec{x}) \leq 0 \quad i=1,2,\cdots,m$ *Less than inequality constraints.*

$h_i(\vec{x}) = 0 \quad i=1,2,\cdots,p$ *Equality constraints.*

$x_{iL} \leq x_i \leq x_{iU} \; i=1,2,\cdots,n$ *Bounds on optimization variables*

- Classification:
  - Unconstrained problems: Minimize (maximize) the (nonlinear) objective function.
  - Linear programming problems: The objective function and all the constraints are linear.
  - Quadratic programming problems: The objective function is quadratic and the constraints linear.
  - Nonlinear programming problems: This is the general optimization problem. One or more constraints are nonlinear.

A graphical example in two variables.

- Minimize the objective function $f(x_1,x_2)$ such that:

$$g_i(x_{1,}x_2)\le 0 \quad i=1,2,\cdots,m$$
$$h_i(x_{1,}x_2)=0 \quad i=1,2,\cdots,p$$

- Procedure:
  - Choose an appropriate range for x_1,x_2 .
  - Draw constraints (or the border).
  - Draw $f(x_1,x_2)$ contour levels.

- Choose an appropriate value for x_1 *or* x_2 if there is information available. If not, choose one arbitrarily.
  - Choose a range for the other variable (Eg.: x_2 if x_1 has been fixed) and solve for the constraints.
- Draw a set of contours for the constraints by choosing a few x_1 values within the allowed range and solving for x_2
- The contours of the objective function are drawn solving o $f(x_1,x_2)=c$ for a choice of c values. As a starting point is usual to use:

$$c_1 = f\left(\frac{1}{3}(x_{1\max} - x_{1\min}), \frac{1}{3}(x_{2\max} - x_{2\min})\right)$$
$$c_2 = f\left(\frac{2}{3}(x_{1\max} - x_{1\min}), \frac{2}{3}(x_{2\max} - x_{2\min})\right)$$

Example:

- Minimize:

$$f(x_{1,} x_2) = 4x_1^2 - 5x_1 x_2 + x_2^2$$

Such that:

$$g(x_{1,} x_2) = x_1^2 - x_2 + 2 \leq 0$$
$$h(x_{1,} x_2) = x_1 + x_2 - 6 = 0$$

- Bounds on the variables:
  - Try with x_1 in *(0,10)*.
  - Solve $g(0,x_2)=0$: $0^2 - x_2 + 2 = 0$ gives $x_2=2$

    Solve $g(10,x_2)=0$ gives $x_2=102$
  - Solve $h(0,x_2)=0$ gives $x_2=6$

    Solve $h(10,x_2)=0$ *gives* $x_2=-4$
- To start we select then x_1 in *(0,10)* and x_2 in

(-4,102)

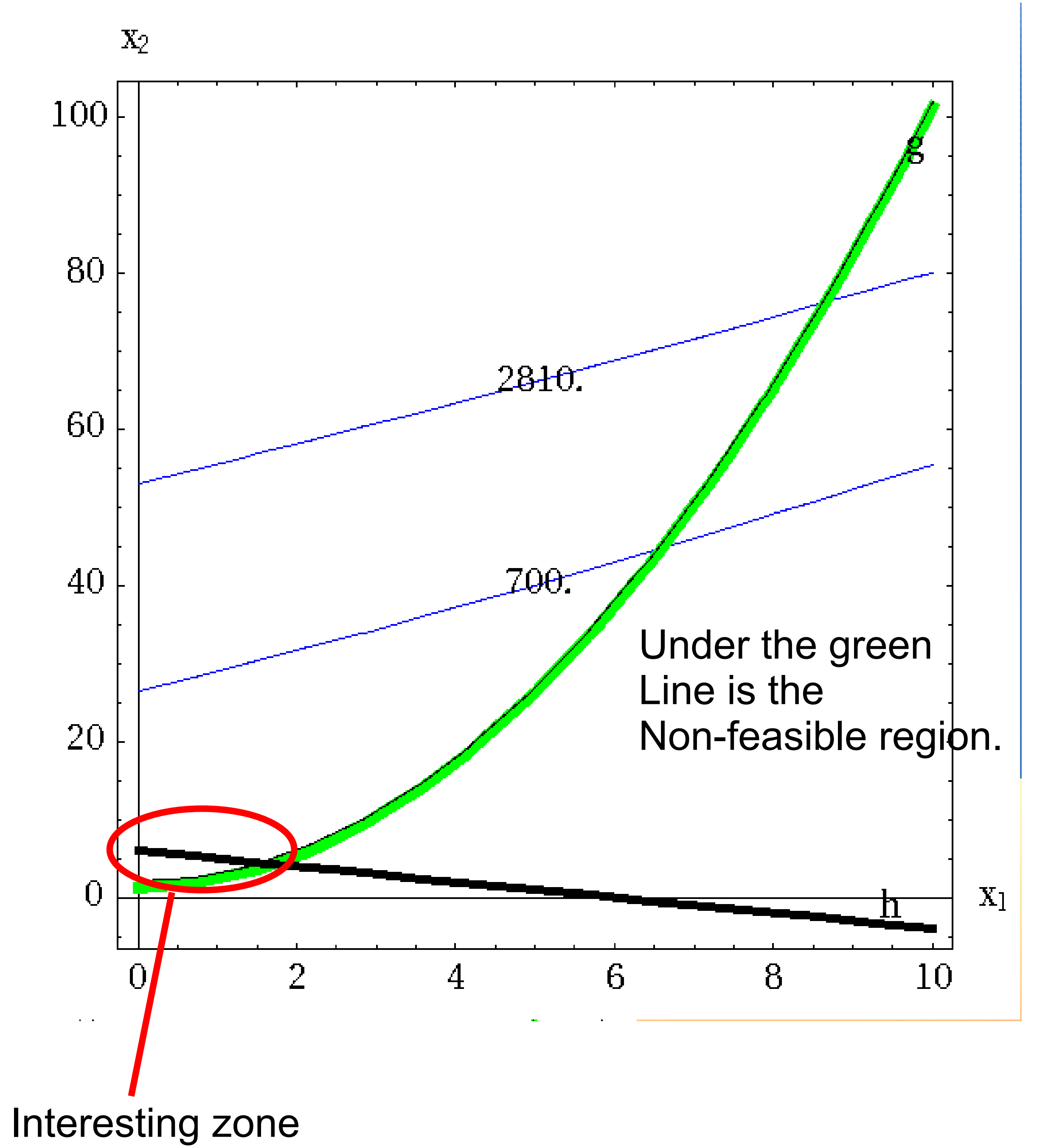
- The contours of *g* and *h* are obtained fixing a set $\{x_{i1}\}$ in *(0,10)* and solving for x_2 g($\{x_{i1}\},x_2$) and h($\{x_{i1}\},x_2$). This produces two sets of points $\{(x_{i1},x_{i2g})\}$ and $\{(x_{i1},x_{i2h})\}$ that define two contours. The one for *h* (equality constraint) defines a line of possible solutions. The one for *g* (inequality constraint) defines a feasible region.

- The contours of the function to minimize are calculated at 1/3 and 2/3 in the interval:

$$c_1 = f\left(\frac{1}{3}(10-0), \frac{1}{3}(102-(-4))\right) = 700$$

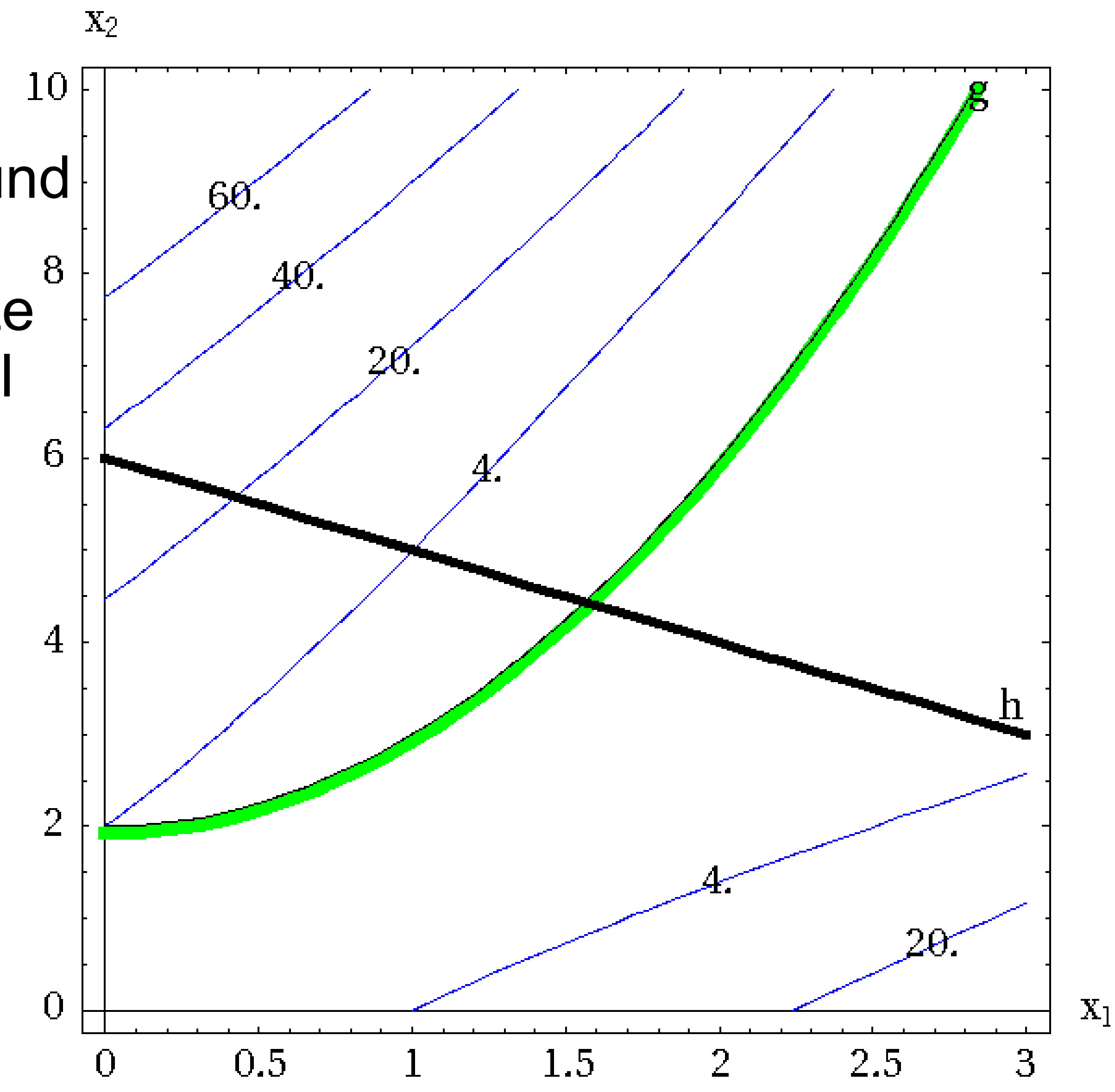
$$c_2 = f\left(\frac{2}{3}10, \frac{2}{3}106\right) = 2810$$

- Drawing everything together:
  - Checking the values *f(1,10)=54*, *f(0,2)=4* we get an idea of function's values around the most interesting zone.

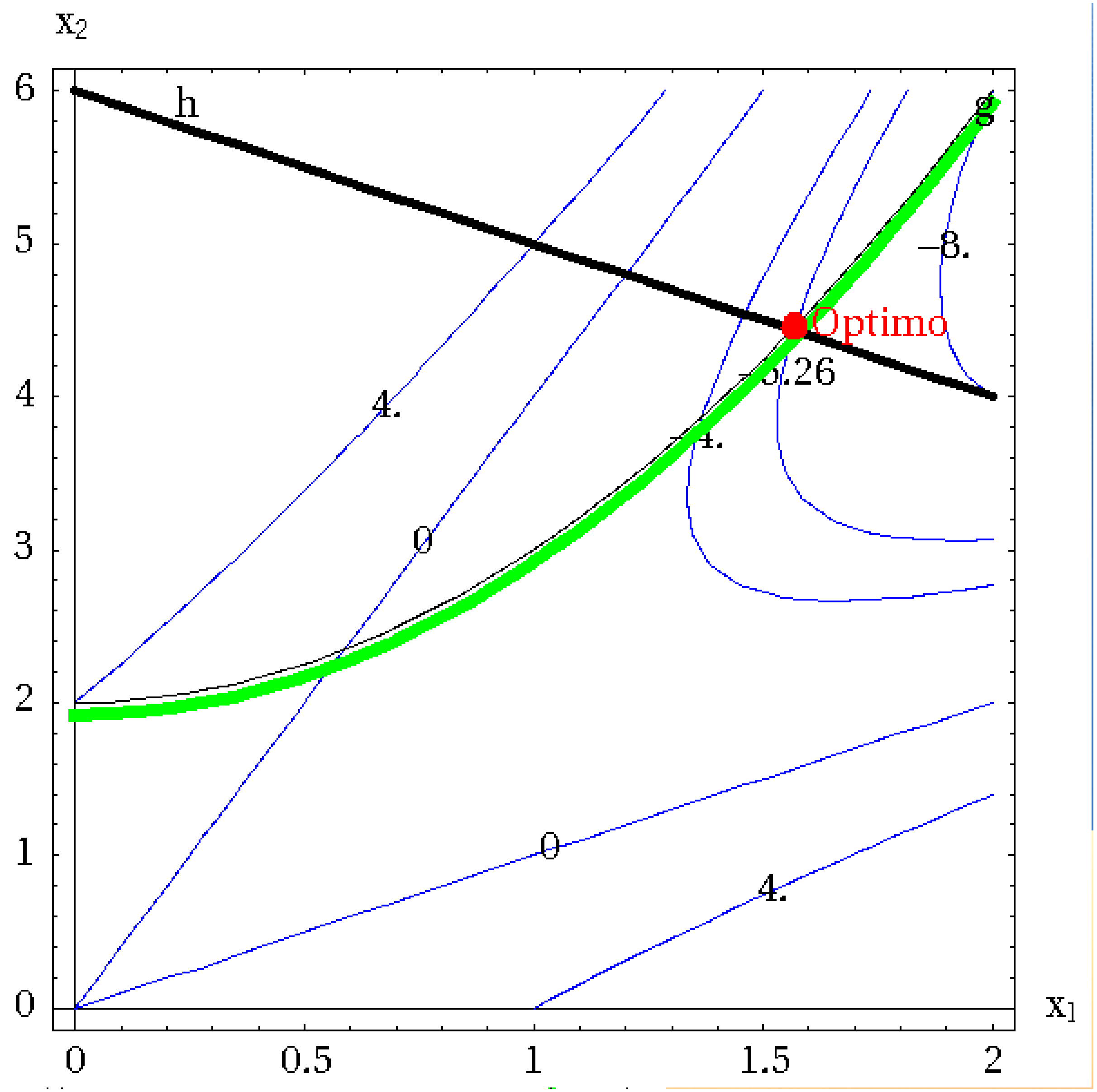


- Zooming in the interesting zone:

- Drawing contours around the estimated values.
- The function to minimize decrease almost parallel to the restriction g.
- Maybe the intersection among g and h is the solution.



- To check the hypotesis, we obtain from the graph the values at the intersection (1.57,4.46), and evaluate (1.57,4.46)=-5.26, Using this value to calculate contours in a close region and see its behavior we see that this is really the minimum satisfying all the constraints.



- Example of a disjoint feasible region.
  - Minimize:

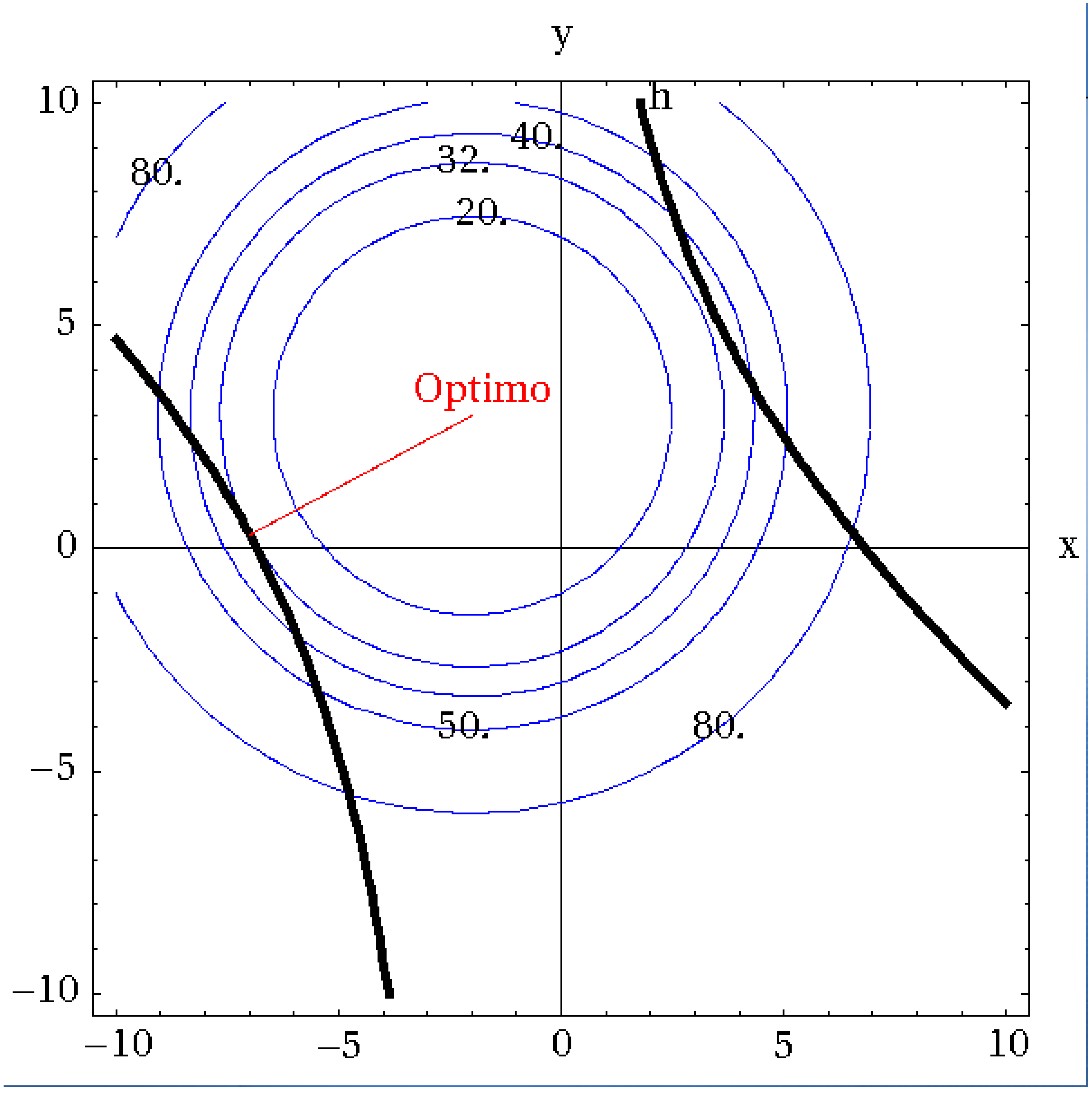
$$f(x, y) = (x+2)^2 + (y-3)^2$$

Such that

$$h(x, y) = 3x^2 + 4xy + 6y = 140$$

The function is a paraboloid of revolution centered in (-2,3).

Optimum in
(-7,0.3)



- Note that a small variation in the function can lead to a very different solutions.
  - Minimize

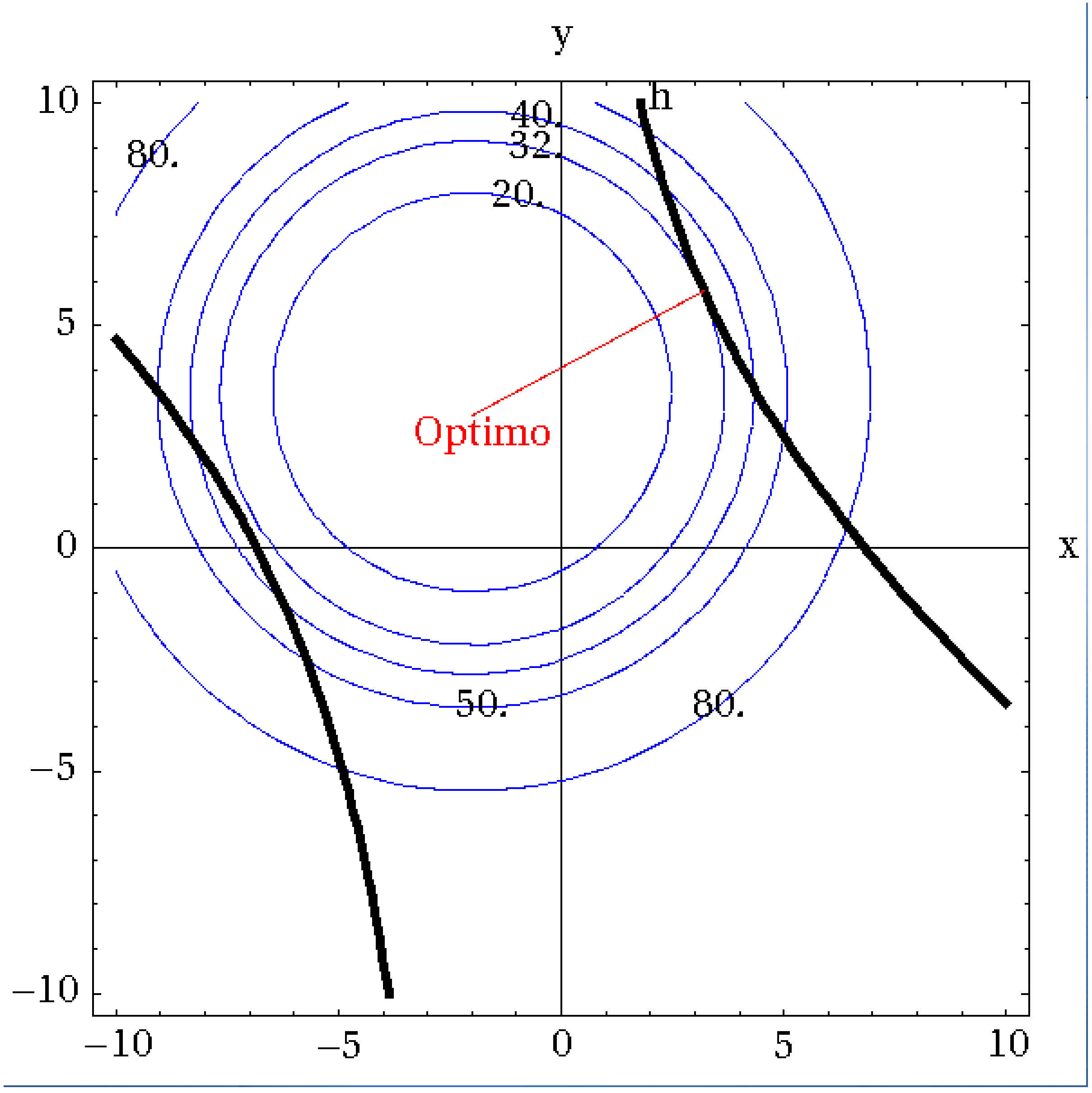
$$f(x, y) = (x+2)^2 + (y-3.5)^2$$

Such that

$$h(x, y) = 3x^2 + 4xy + 6y = 140$$

The function is a paraboloid of revolution centered in (-2,3.5).

Optimum in
(3.2,5.8)

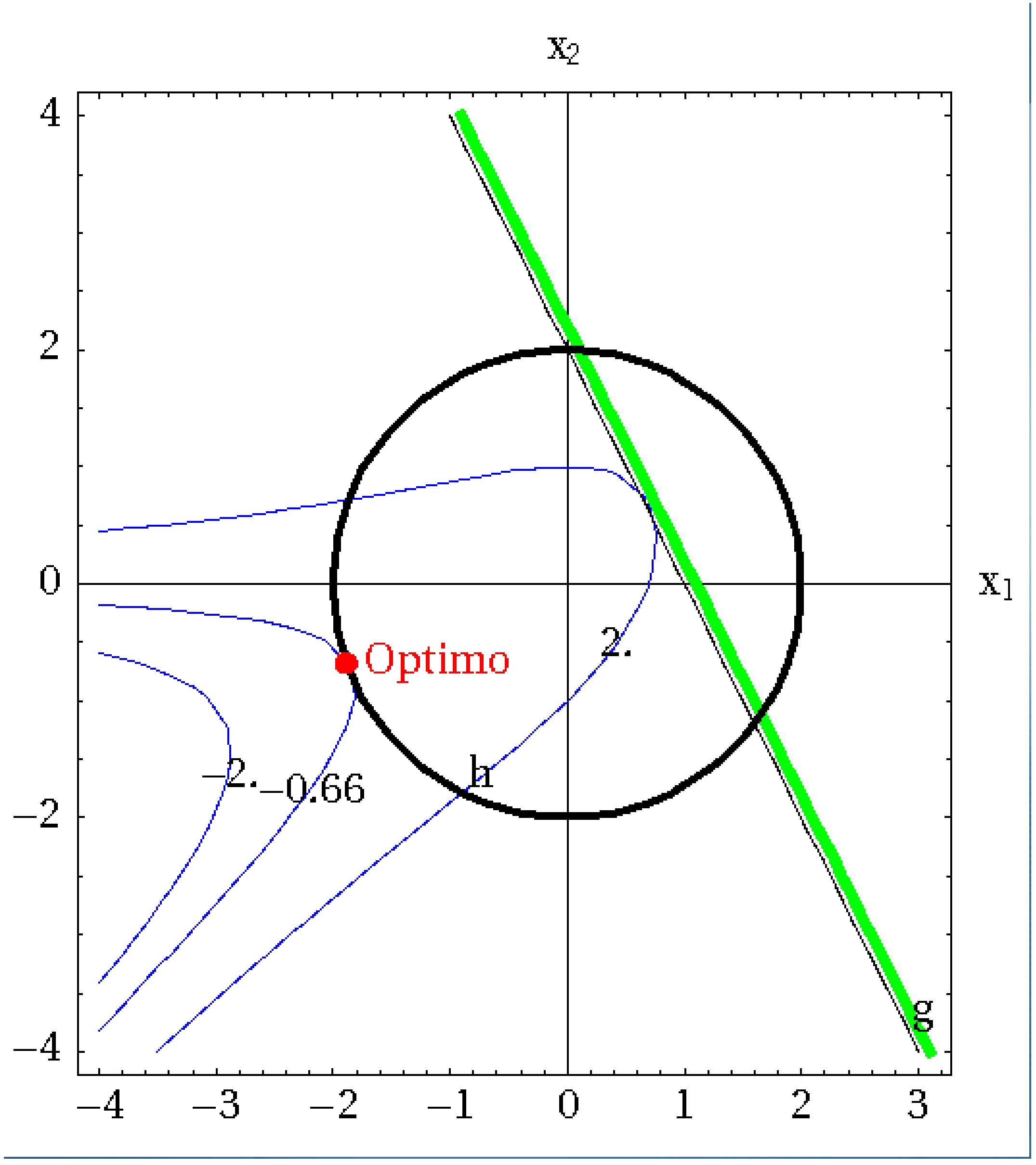


- Try the solution of::
  - Minimize:

$$f(x_{1,} x_2) = \mathrm{e}^{x_1} - x_1 x_2 + x_2^2$$

Such that

$$g(x_{1,} x_2) = 2x_1 + x_2 - 2 \leq 0$$
$$h(x_{1,} x_2) = x_1^2 + x_2^2 = 4$$



x2
x1
Optimo
2.
-2.
-0.66
h
g
4
2
0
-2
-4
-3
-1
1
3

Recap of a few basic mathematical notions.

- Taylor series in the neighborhood of x_0:
  - One variable:

$$f(x) = f(x_0) + \frac{df(x_0)}{dx}(x - x_0) + \frac{1}{2!}\frac{d^2 f(x_0)}{dx^2}(x - x_0)^2 + \cdots$$

$$+ \frac{1}{n!}\frac{d^n f(\xi)}{dx^n}(x - \xi)^n. \qquad \xi \in [x, x_0]$$

– In n variables, around $\vec{x}_0 = (x_{10}, x_{20}, \cdots, x_{n0})^T$

$$f(\vec{x}) = f(\vec{x}_0) + \nabla f(\vec{x}_0)(\vec{x} - \vec{x}_0) + \frac{1}{2!}(\vec{x} - \vec{x}_0)^T \nabla^2 f(\vec{x}_0)(\vec{x} - \vec{x}_0) + \cdots$$

$$\nabla f(\vec{x}) = \begin{pmatrix} \dfrac{\partial f(\vec{x})}{\partial x_1} \\ \dfrac{\partial f(\vec{x})}{\partial x_2} \\ \vdots \\ \dfrac{\partial f(\vec{x})}{\partial x_n} \end{pmatrix} \qquad \nabla^2 f(\vec{x}) = \begin{pmatrix} \dfrac{\partial^2 f(\vec{x})}{\partial x_1^2} & \dfrac{\partial^2 f(\vec{x})}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f(\vec{x})}{\partial x_1 \partial x_n} \\ \dfrac{\partial f(\vec{x})}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f(\vec{x})}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f(\vec{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial^2 f(\vec{x})}{\partial x_n \partial x_1} & \dfrac{\partial^2 f(\vec{x})}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f(\vec{x})}{\partial x_n^2} \end{pmatrix}$$

Gradient vector

Hessian matrix. Symmetric if $f \in C^2$

- The gradient vector $\nabla f(\vec{x})$ is perpendicular to the (hyper)surface defined by $f(\vec{x}) = cte.$
  - First order approximation:

$$f(\vec{x}) \sim f(\vec{x}_0) + (\vec{x} - \vec{x}_0)^T \nabla f(\vec{x}_0)$$

If $\vec{x}_0$ is on the surface $f(\vec{x}) - f(\vec{x}_0) = 0$

hence:

$$(\vec{x} - \vec{x}_0)^T \nabla f(\vec{x}_0) = 0$$

That defines the plane normal to the surface in $\vec{x}_0$

- The steepest descent direction is minus the gradient.

- Non linear equations systems: Multidimensional Newton-Raphson.
  - Solve the system: $\vec{f}(\vec{x})=\vec{0}$

$$
\begin{gathered}
f_1(x_{1,}x_{2,}\cdots,x_n)=0\\
f_2(x_{1,}x_{2,}\cdots,x_n)=0\\
\vdots\\
f_n(x_{1,}x_{2,}\cdots,x_n)=0
\end{gathered}
$$

Using an iterative method $\vec{x}^{k+1}=\vec{x}^k+\Delta\vec{x}^k$ where the value for the correction is calculated using a first order Taylor series:

$$
\begin{gathered}
f_1(\vec{x}^{k+1})=f_1(\vec{x}^k)+\nabla f_1(\vec{x}^k)^T\Delta\vec{x}^k\\
f_2(\vec{x}^{k+1})=f_2(\vec{x}^k)+\nabla f_2(\vec{x}^k)^T\Delta\vec{x}^k\\
\vdots
\end{gathered}
$$

– In matrix form:

$$\begin{pmatrix} f_1(\vec{x}^k) \\ f_2(\vec{x}^k) \\ \vdots \\ f_n(\vec{x}^k) \end{pmatrix} + \begin{pmatrix} \dfrac{\partial f_1(\vec{x}^k)}{\partial x_1} & \dfrac{\partial f_1(\vec{x}^k)}{\partial x_2} & \cdots & \dfrac{\partial f_1(\vec{x}^k)}{\partial x_n} \\ \dfrac{\partial f_2(\vec{x}^k)}{\partial x_1} & \dfrac{\partial f_2(\vec{x}^k)}{\partial x_2} & \cdots & \dfrac{\partial f_2(\vec{x}^k)}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial f_n(\vec{x}^k)}{\partial x_1} & \dfrac{\partial f_n(\vec{x}^k)}{\partial x_2} & \cdots & \dfrac{\partial f_n(\vec{x}^k)}{\partial x_n} \end{pmatrix} \Delta \vec{x}^k = \vec{0}$$

$$\vec{f}(\vec{x}^k) + \boldsymbol{J}(\vec{x}^k) \Delta \vec{x}^k = \vec{0}$$
$$\Delta \vec{x}^k = -\boldsymbol{J}^{-1}(\vec{x}^k) \vec{f}(\vec{x}^k)$$

J is the Jacobian matrix.

- Quadratic Forms:
  - A quadratic form is a function of n variables where each term is the square of a variable or the product of two.
  - A quadratic form can be written using a symmetric matrix ***A*** as:

$$f(\vec{x}) = \frac{1}{2}\vec{x}^T \boldsymbol{A}\vec{x}$$

Gradient and Hessian: $\nabla f(\vec{x}) = \boldsymbol{A}\vec{x}$, $\nabla^2 f(\vec{x}) = \boldsymbol{A}$

- Ej: $f(x_1, x_2, x_3) = x_1^2 + x_1 x_2 - 2x_1 x_3 - 4x_2^2 - x_2 x_3 + 3x_3^2$

  Is a quadratic form with matrix ***A*** :

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 & -2 \\ 1 & -8 & -1 \\ -2 & -1 & 6 \end{pmatrix}$$

- Convexity is an important property of optimization problems. It is determined by looking the *definition* properties of the Hessian matrix.
  - To check this properties, the ***minors*** of the matrix are used.
  - The minors are the determinants of the square submatrices of the matrix.
  - The first principal minor is the first diagonal element (submatrix 1×1). The second is the value of the determinant of the 2×2 submatrix formed by the two first rows and two first columns, etc.

- The symmetric matrix ***A***, with minors A_i, is said:

  a) Positive definite if $A_i > 0$; i=1,...n

  b) Positive semidefinite if $A_i >= 0$; i=1,...n

  c) Negative definite if $(-1)^i A_i > 0$; i=1,...n

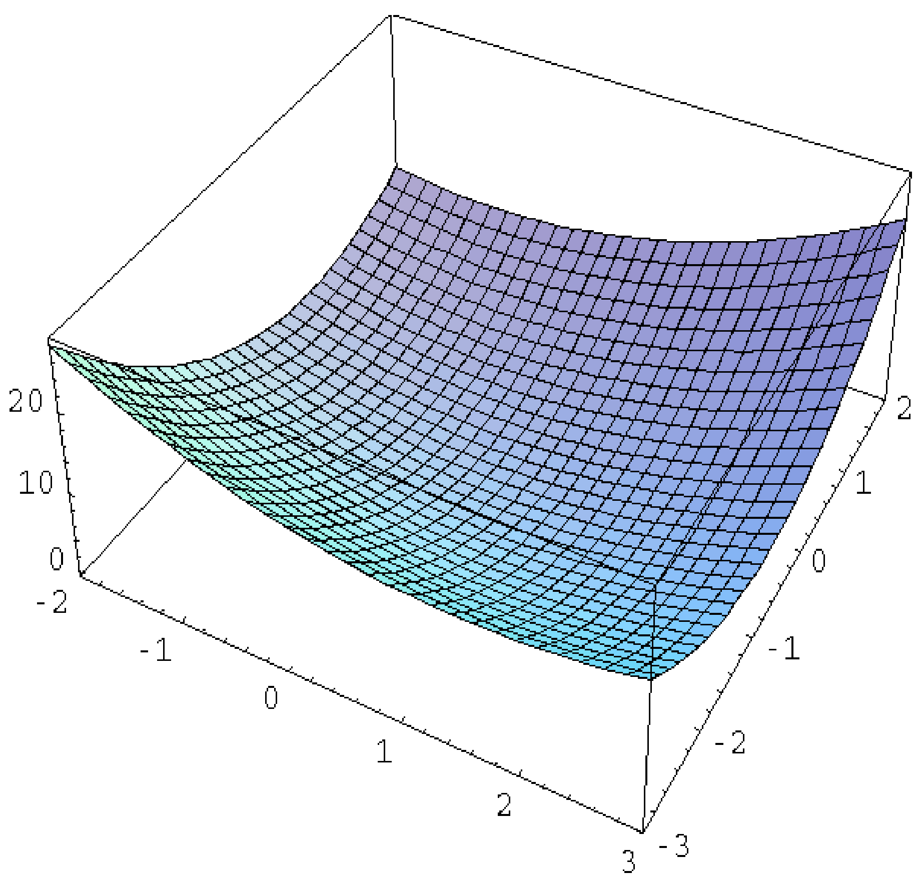
  d) Negative semidefinite if $(-1)^i A_i >= 0$; i=1,...n

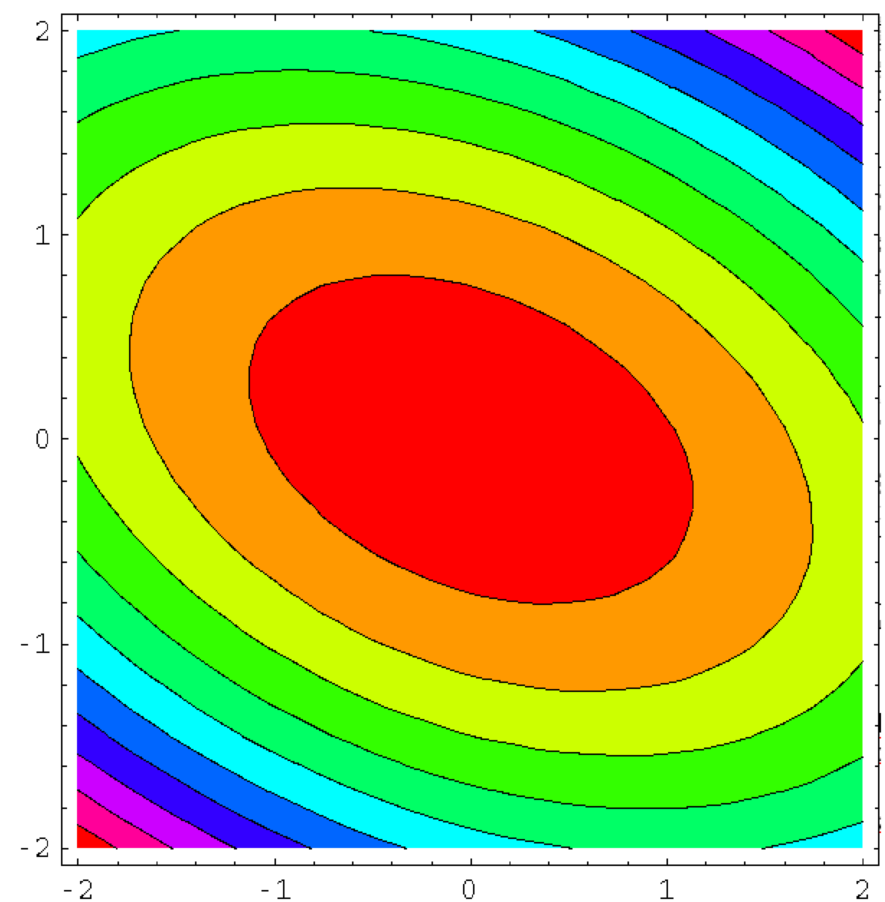
  e) Indefinite, if none of the previous cases applies.

- Since every symmetric matrix can be associated to a quadratic form, we can see graphically the meaning of the aforementioned properties using two variables:

$$A=\begin{pmatrix}2 & 1\\ 1 & 4\end{pmatrix} \quad \textit{The associated cuadratic form is} f(x,y)=x^2+xy+2y^2$$

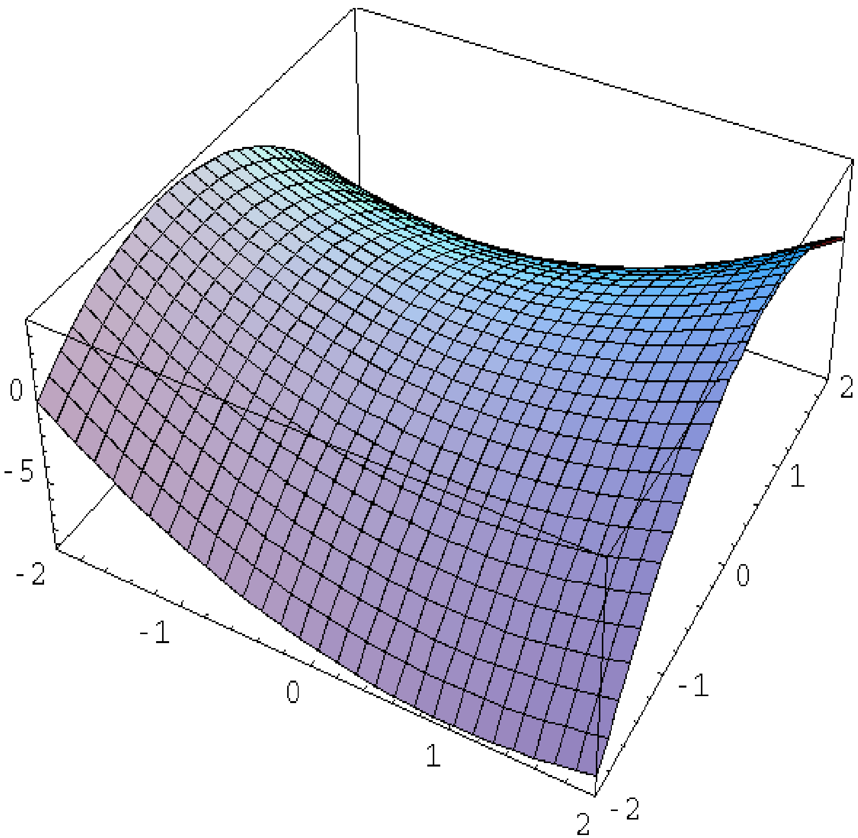
Its minors are 2,7, *hence it is* ***positive definite***

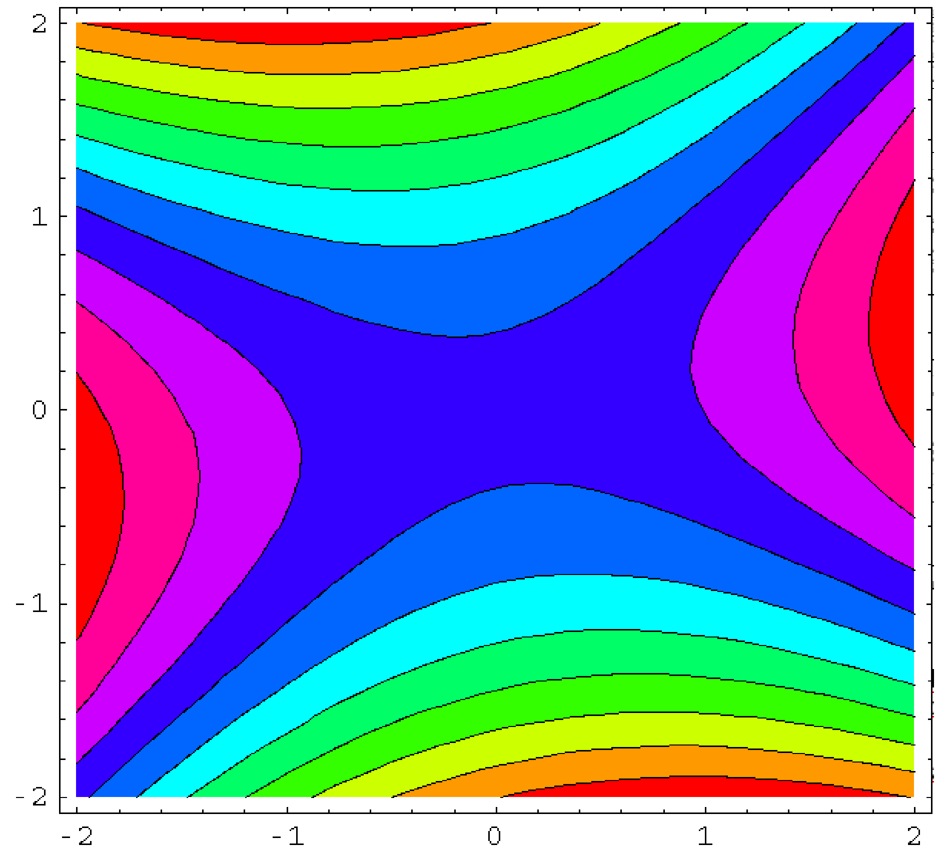




$$A=\begin{pmatrix}2 & 1\\ 1 & -4\end{pmatrix}$$ *The associated cuadratic form is* $f(x,y)=x^2+xy-2y^2$

Its minors are $2,-9$, *hence it is* ***indefinite***

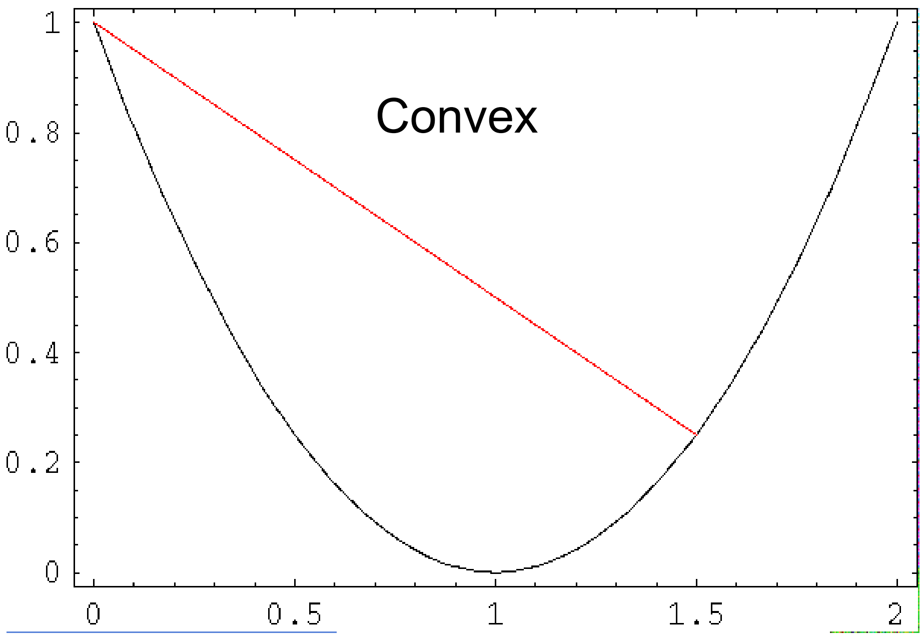


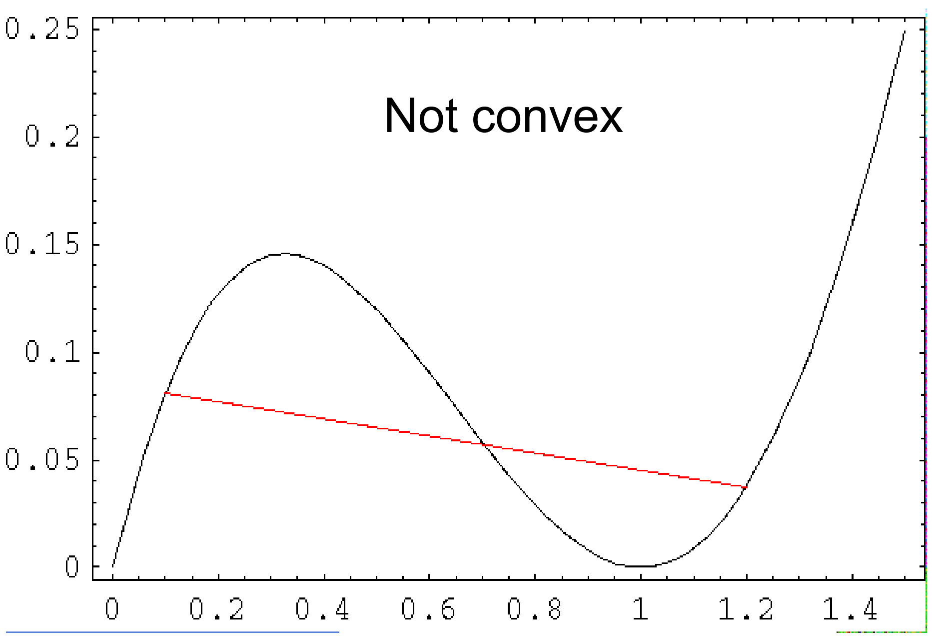


- Functions and convex sets:
  - A function is said convex if for any two points $\vec{x}^{(1)},\ \vec{x}^{(2)}$ is satisfied:

$$f\left(\alpha\vec{x}^{(2)}+(1-\alpha)\vec{x}^{(1)}\right)\leq\alpha f\left(\vec{x}^{(2)}\right)+(1-\alpha)f\left(\vec{x}^{(1)}\right)$$

    - Graphically: For any two points of the function, the graph of the function is always under the straight line joining the two points.





- A set is convex if for any two points $\vec{x}^{(1)}$, $\vec{x}^{(2)}$ in the set, then:

$$\vec{x} = \alpha \vec{x}^{(2)} + (1-\alpha)\vec{x}^{(1)} \quad 0 \le \alpha \le 1$$

Is also in the set.

  - The lines joining any two points in the set belong completely in the set.

  - **NOTATION:** boldface letters will be used to denote vectors. $\vec{x} \equiv \boldsymbol{x}$

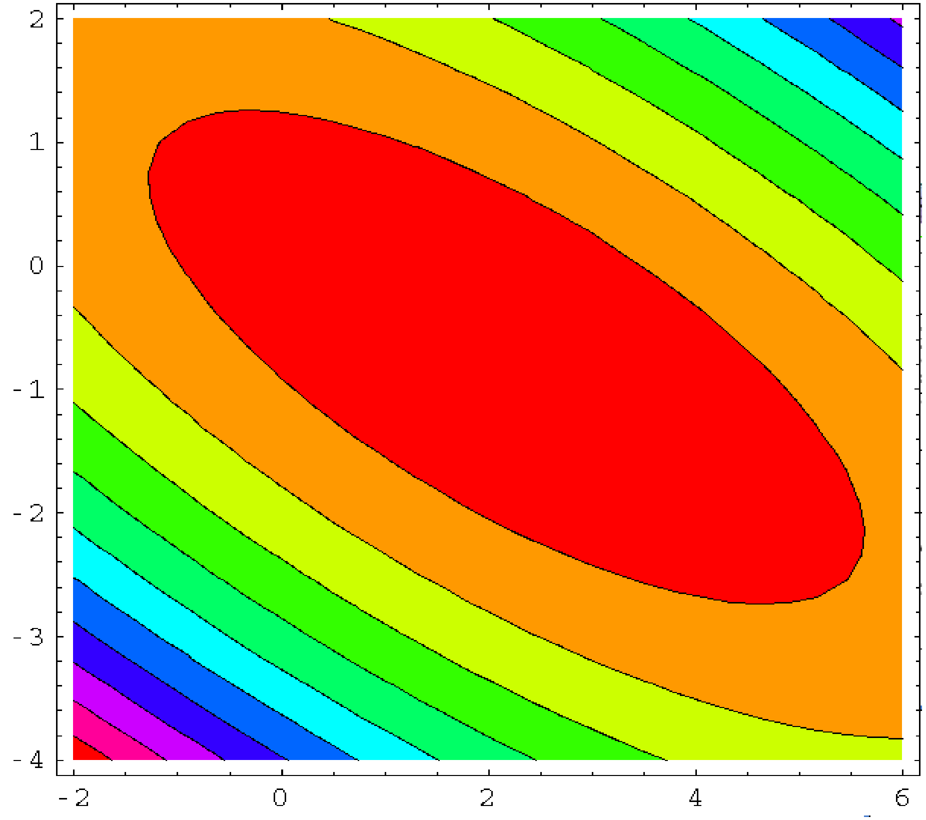
- How to find out if a function is convex?
  - A function is convex if its Hessian is at least positive semidefinite.
  - A function is concave if its Hessian is at least negative semidefinite.
  - A function is convex indefinite if its Hessian is indefinite.
  - Moreover:
    - *If* $f(x)$ *is convex* $\alpha f(x)$ *is convex* $\forall \alpha > 0$
    - *The sum of convex functions is convex.*
    - *If f(x) is convex and g(y) is growing and convex, then g(f(x)) is also convex.*

- Examples:

$$f(x,y)=5-5x-2y+2x^2+5xy+6y^2$$

$$\textit{Hessian}: \ \nabla^2 f(x,y)=\begin{pmatrix}4 & 5\\ 5 & 12\end{pmatrix}$$

$$\textit{Minors}: \ A_1=4; \ \ A_2=23; \ \rightarrow \ \textit{Positive definite} \ \rightarrow \ \textit{Convex}$$



$$f(x, y) = \mathrm{e}^{x^2+y^2} + \mathrm{e}^{x+2y}$$

$$\nabla^2 f = \begin{pmatrix} 4x^2 \mathrm{e}^{x^2+y^2} + \mathrm{e}^{x+2y} + 2\mathrm{e}^{x^2+y^2} & 4xy\,\mathrm{e}^{x^2+y^2} + 2\mathrm{e}^{x+2y} \\ 4xy\,\mathrm{e}^{x^2+y^2} + 2\mathrm{e}^{x+2y} & 4y^2 \mathrm{e}^{x^2+y^2} + 4\mathrm{e}^{x+2y} + 2\mathrm{e}^{x^2+y^2} \end{pmatrix}$$

To explicitly test this Hessian is tedious. We can apply the rules for Composing convex functions:

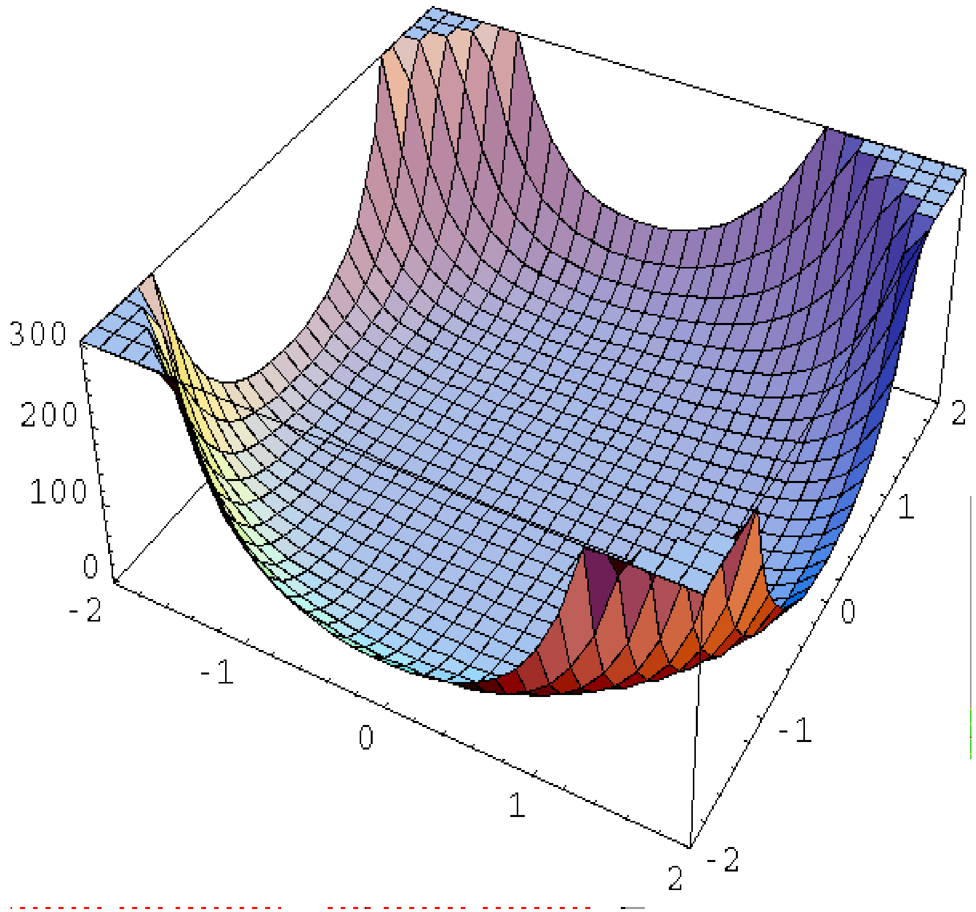
- $x^2 + y^2$ Is convex (a paraboloid of revolution). We can test this. Its Hessian is:

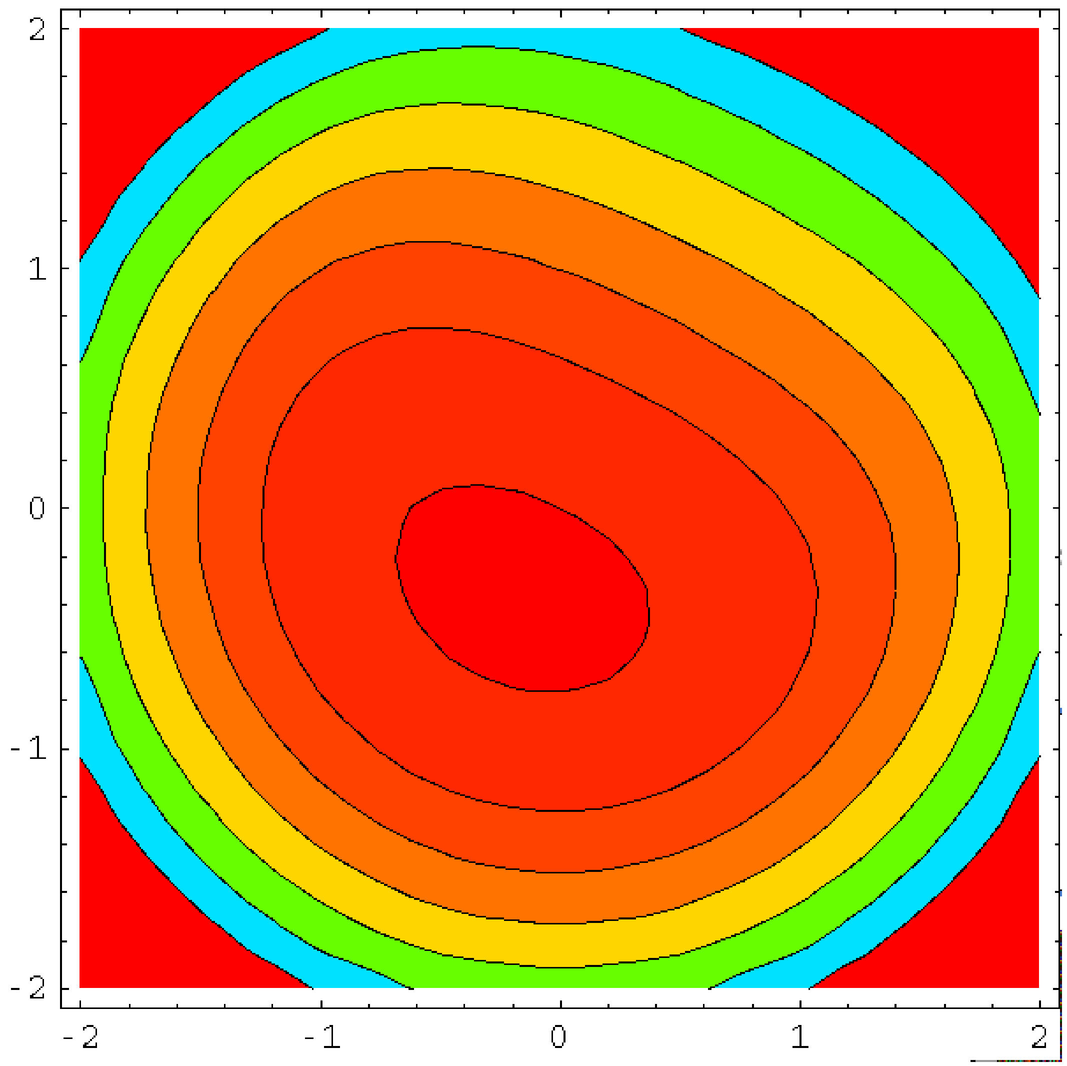
$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \; \textit{with minors} \;\; A_1 = 2; \;\; A_2 = 4 \;\; \rightarrow \;\; \textit{positive definite} \;\; \rightarrow \;\; \textit{convex}$$

- $x + 2y$ Is a linear function, hence trivially convex.

- e^x Is a growing function, hence $\mathrm{e}^{x^2+y^2} \wedge \mathrm{e}^{x+2y}$ are convex, and also its sum.

$$f(x, y) = e^{x^2+y^2} + e^{x+2y}$$





- The convex optimization problem.
  - An optimization problem where the objective function is convex and the feasibility region is a convex set is called the “convex optimization problem”.
  - In a convex optimization problem, the minimum is global.
    - A linear programming problem is always convex.
    - A non linear programming problem with convex objective function with linear constraints of any kind or non linear convex **inequality** constraints is convex.
      - If any of the nonlinear constraints is an equality, no matter if convex, the problem will be non convex.

- In a convex optimization problem, the minimum is global.

If $\boldsymbol{x}_l$ *is not the global minima, there will be a* $\boldsymbol{x}_g$ *such that*:

$$f(\boldsymbol{x}_g) < f(\boldsymbol{x}_l)$$

The points in *the connecting line* :

$$\boldsymbol{x} = \alpha\boldsymbol{x}_g + (1-\alpha)\boldsymbol{x}_l \quad 0 \le \alpha \le 1$$

since they are a feasible convex set, they will be in *the feasible set. Since* $f(x)$ *is convex, then*:

$$f(\boldsymbol{x}) \le \alpha f(\boldsymbol{x}_g) + (1-\alpha) f(\boldsymbol{x}_l)$$

$$f(\boldsymbol{x}) \le f(\boldsymbol{x}_l) + \alpha\left(f(\boldsymbol{x}_g) - f(\boldsymbol{x}_l)\right)$$

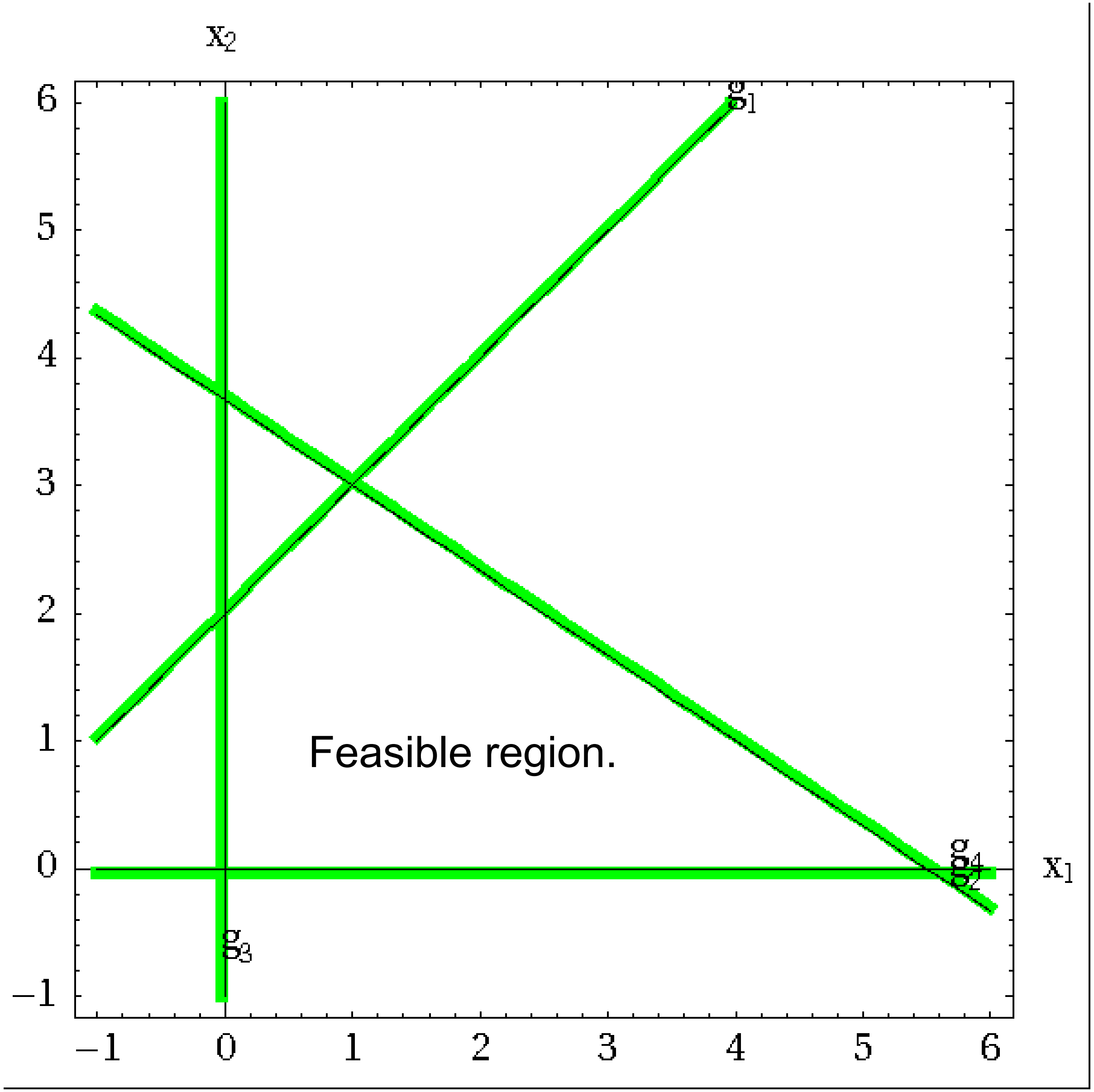
Since $f(\boldsymbol{x}_g) - f(\boldsymbol{x}_l)$ *is negative, then*:

$$f(\boldsymbol{x}) \le f(\boldsymbol{x}_l) \quad \forall \;\; 0 \le \alpha \le 1$$

• Examples:

$$g_1 = -x + y \leq 2$$
$$g_2 = 2x + 3y \leq 11$$
$$g_3 = x \geq 0$$
$$g_4 = y \geq 0$$

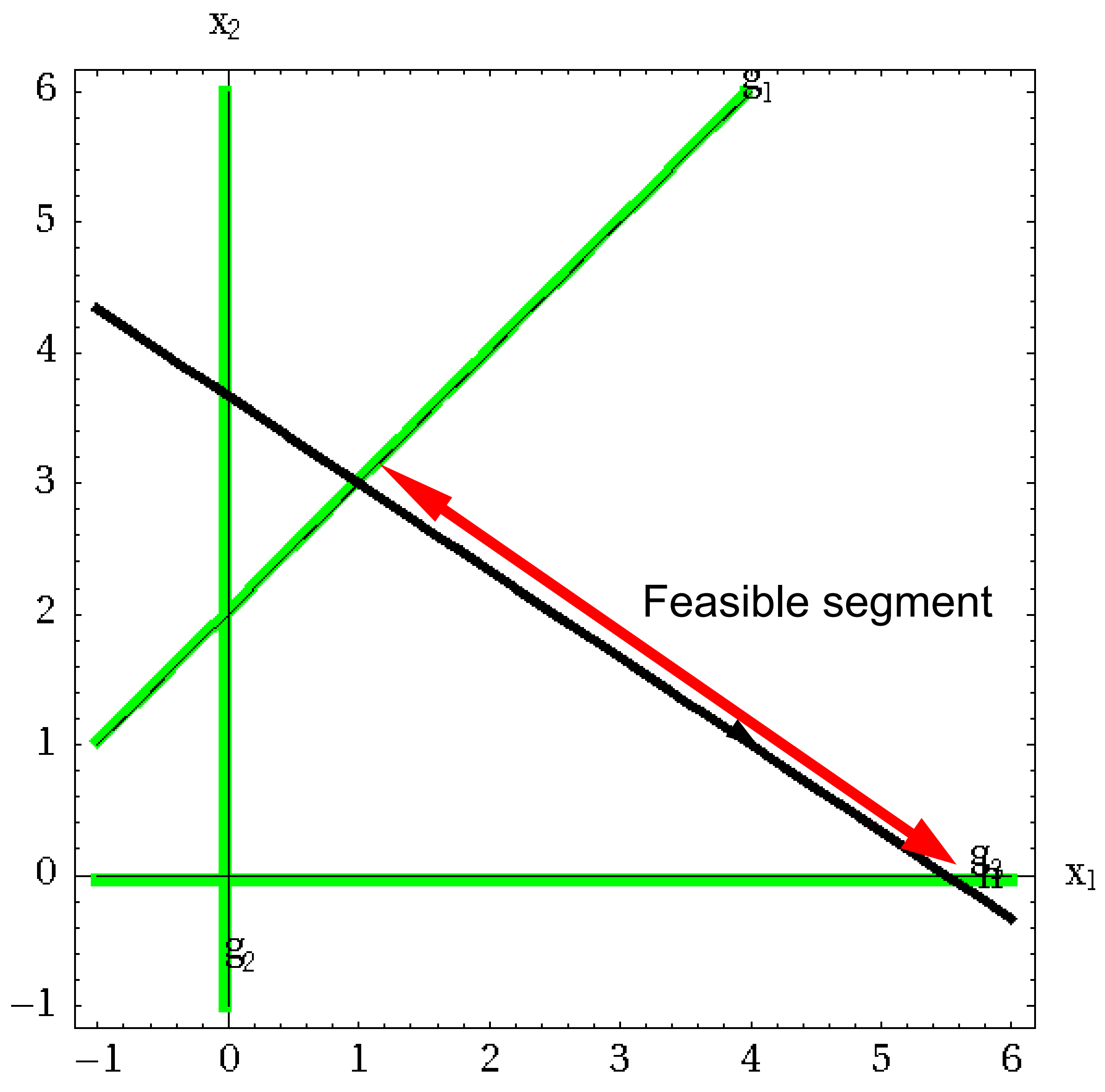
Linear inequality Constraints = Convex feasibility region.



- Examples:

$$g_1 = -x + y \leq 2$$
$$g_2 = 2x + 3y = 11$$
$$g_3 = x \geq 0$$
$$g_4 = y \geq 0$$

Linear equality or inequality Constraints = Convex feasibility Region (a straight line segment)



- Examples:

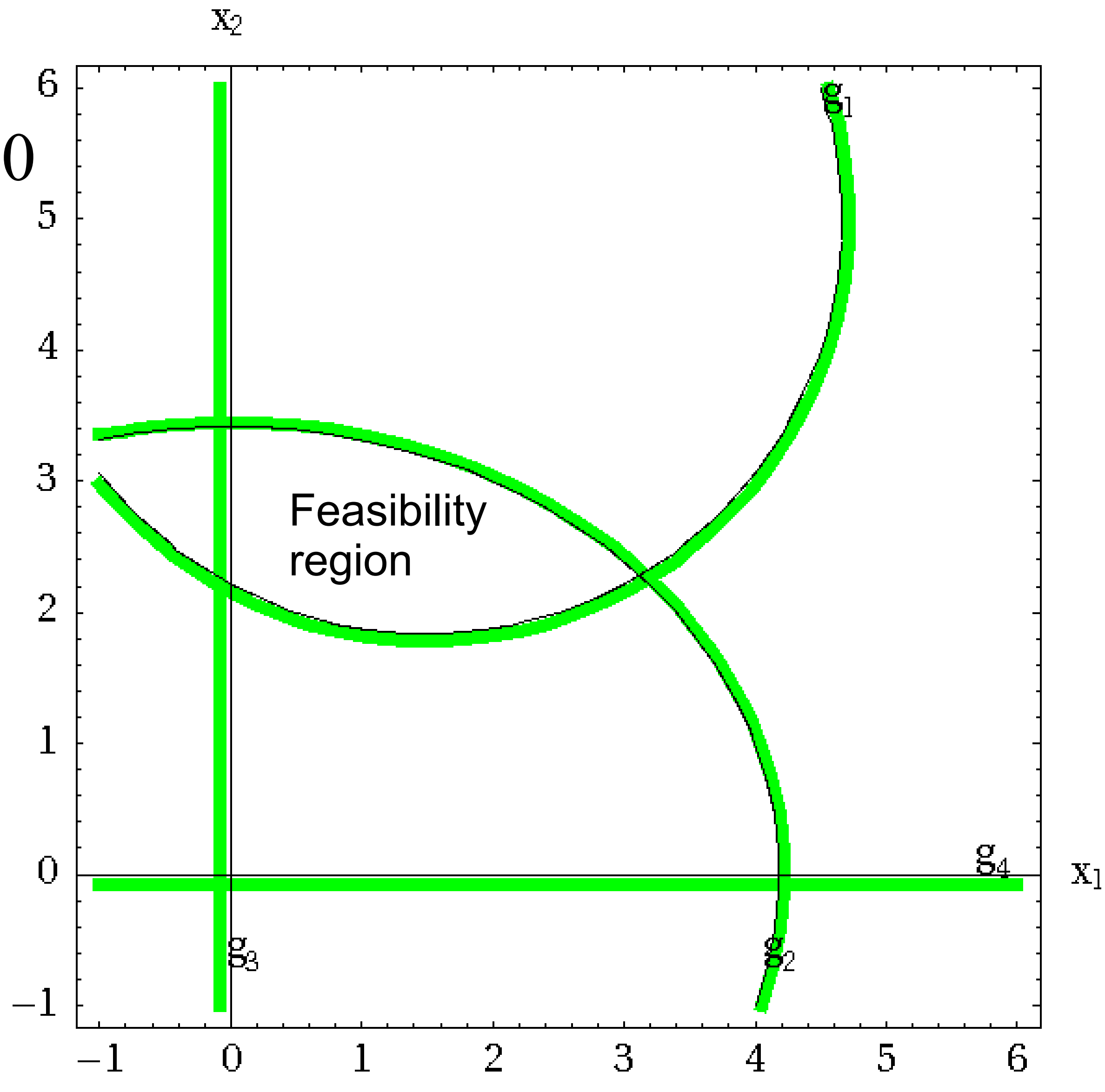
$$g_1 = (x - \frac{3}{2})^2 + (y-5)^2 \le 10$$

$$g_2 = 2x^2 + 3y^2 \le 35$$

$$g_3 = x \ge 0$$

$$g_4 = y \ge 0$$

Convex inequality, linear or not, constraints = Convex feasibility region.



- Examples:

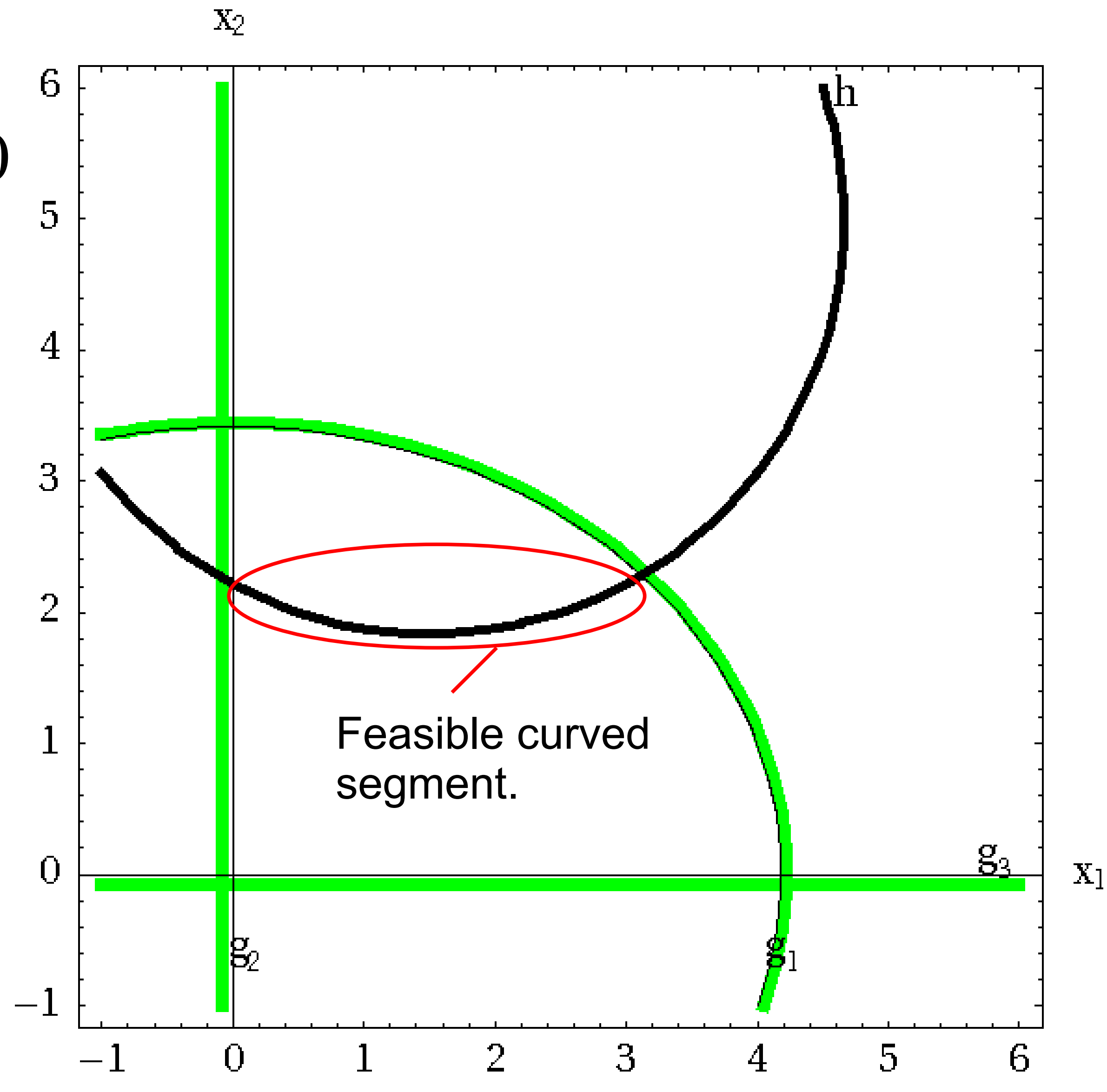
$$h = \left(x - \frac{3}{2}\right)^2 + (y-5)^2 = 10$$

$$g_2 = 2x^2 + 3y^2 \leq 35$$

$$g_3 = x \geq 0$$

$$g_4 = y \geq 0$$

Inequality and equality nonlinear convex constraints = nonconvex feasibility region: A segment connecting two points of the feasible segment has points outside of the feasibility region (segment)



Optimality Conditions. Unconstrained Problems: Extension to more than one variable.

- Necessary Condition:
  - A Taylor series around the optimum: $\boldsymbol{x}^*$

$$f(\boldsymbol{x}) \sim f(\boldsymbol{x}^*) + (\boldsymbol{x} - \boldsymbol{x}^*)^T \nabla f(\boldsymbol{x}^*)$$

or

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) = (\boldsymbol{x} - \boldsymbol{x}^*)^T \nabla f(\boldsymbol{x}^*)$$

the first term is always bigger or *equal* to *zero.*

$(\boldsymbol{x} - \boldsymbol{x}^*)^T$ *can ber either positive* or *negative*, *hence* to *first order*, to *satisfy the equality*:

$$\nabla f(\boldsymbol{x}^*) = 0$$

- Note: This is a “zero slope” condition that is also satisfied in a maximum or inflection point: It is a necessary condition but not sufficient.

- Sufficient Condition:
  - The Taylor series to second order:

$$f(\boldsymbol{x})-f(\boldsymbol{x}^*)=(\boldsymbol{x}-\boldsymbol{x}^*)^T\nabla f(\boldsymbol{x}^*)+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^T\nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{x}-\boldsymbol{x}^*)$$

since $f(\boldsymbol{x})-f(\boldsymbol{x}^*)\geq 0$ and, *because of the first order* $\nabla f(\boldsymbol{x}^*)=0$ *condition, we have*:

$$(\boldsymbol{x}-\boldsymbol{x}^*)^T\nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{x}-\boldsymbol{x}^*)\geq 0$$

  - This is a cuadratic form. It will always be positive if its associated matrix is positive definite.
    - If it is positive semidefinite, the condition will only be a necessary condition and we will have to go to the next order.
    - If it is negative definite we have a maximum.

- So, we have the following rules in stationary points: $\left(\nabla f(\boldsymbol{x}^{*})=0\right)$
  - If the Hessian matrix in that point is positive definite, is, at least, a local minimum.
  - If the Hessian is negative definite there is, at least, a local maximum.
  - If it is indefinite, it is an inflection point.
  - If it is positive or negative semidefinite, the conditions are not enough and we have to go to a higher order series.

- Example:

$$f(x,y)=25x^2-12x^4-6xy+25y^2-24x^2y^2-12y^4$$

  - The stationary points: $\nabla f(x,y)=0$

$$\frac{\partial f}{\partial x}=50x-48x^3-6y-48xy^2=0$$
$$\frac{\partial f}{\partial y}=-6x+50y-48x^2y-48y^3=0$$

Its solutions are:

$$s_1=(-0.7637, 0.7637)$$
$$s_2=(-0.6770, -0.6770)$$
$$s_3=(0,0)$$
$$s_4=(0.6770, 0.6770)$$
$$s_5=(0.7637, -0.7637)$$

– The Hessian matrix:

$$\nabla^2 f(x,y)=\begin{pmatrix} 50-144x^2-48y^2 & -6-96xy \\ -6-96xy & 50-48x^2-144\mathrm{y}^2 \end{pmatrix}$$

- Evaluated in s_1

$$\begin{pmatrix} -62 & 50 \\ 50 & -62 \end{pmatrix} \textit{ with minors } A_1=-62 \text{ and } A_2=1344$$

$$\textit{Is negative definite} \rightarrow \textit{Maximum}$$

- Evaluated in s_2

$$\begin{pmatrix} -38 & -50 \\ -50 & -38 \end{pmatrix} \textit{ with minors } A_1=-38 \text{ and } A_2=-1056$$

$$\textit{Is indefinite} \rightarrow \textit{Inflection point}$$

- Evaluated in s_3

$$\begin{pmatrix} 50 & -6 \\ -6 & 50 \end{pmatrix} \textit{ with minors } A_1 = 50 \text{ and } A_2 = 2464$$

$$\textit{Is positive definite} \rightarrow \textit{Minimum}$$

- Evaluated in s_4

$$\begin{pmatrix} -38 & -50 \\ -50 & -38 \end{pmatrix} \textit{ with minors } A_1 = -38 \text{ and } A_2 = -1056$$

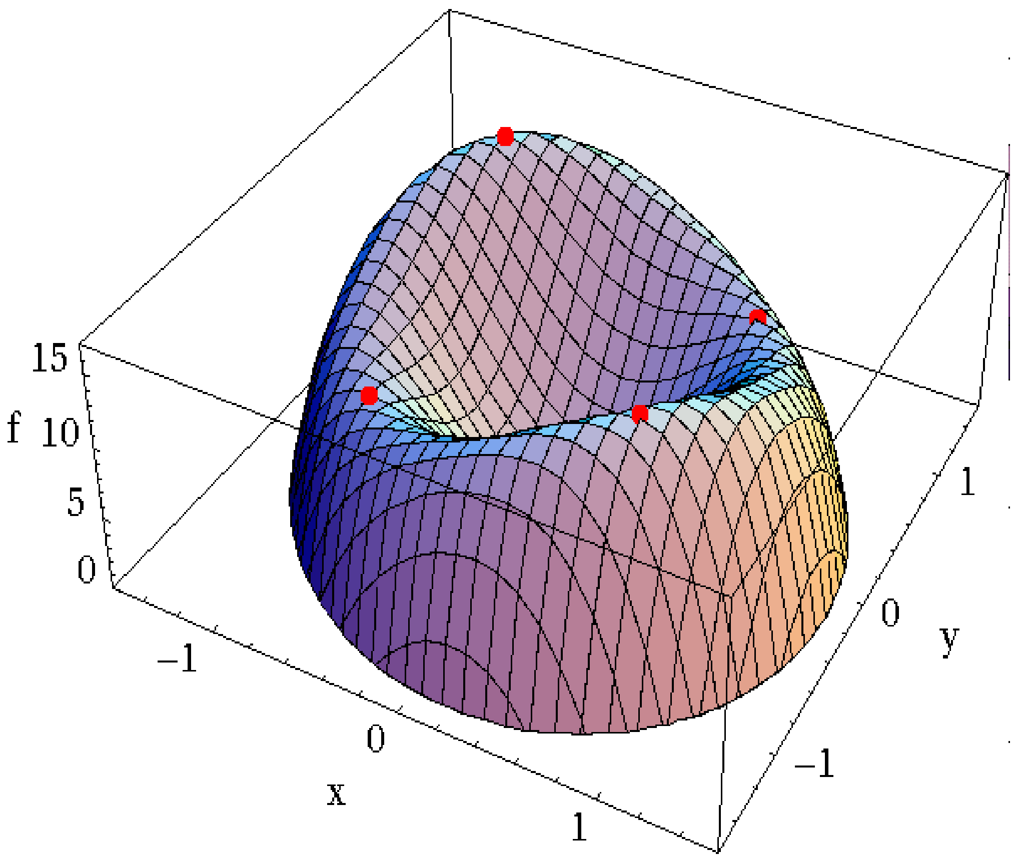
$$\textit{Is indefinite} \rightarrow \textit{Inflection point}$$

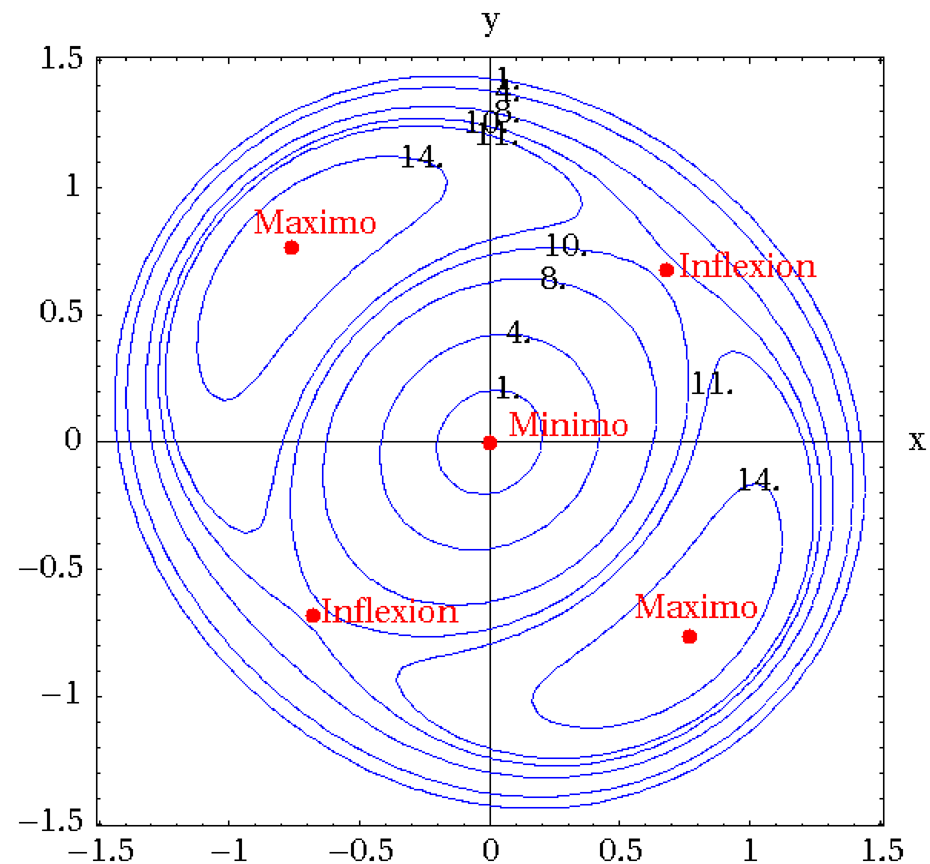
- Evaluated in s_5

$$\begin{pmatrix} -62 & 50 \\ 50 & -62 \end{pmatrix} \textit{ with minors } A_1 = -62 \text{ and } A_2 = 1344$$

$$\textit{Is negative definite} \rightarrow \textit{Maximum}$$

- Graphically:





Optimality Conditions. Constrained Problems.

- Constraint additivity:
  - If a point $\boldsymbol{x}$ satisfies the constraints:

$$
\begin{aligned}
g_i(\boldsymbol{x}) &\leq 0 \quad i=1,\cdots m \\
h_i(\boldsymbol{x}) &= 0 \quad i=1,\cdots p
\end{aligned}
$$

It also satisfies:

$$g_a = \sum_{i=1}^{m} u_i g_i(\boldsymbol{x}) + \sum_{i=1}^{p} v_i h_i(\boldsymbol{x})$$

with $u_i \geq 0, v_i$ *arbitrary scalars*

- First order necessary conditions for the constrained minimization problem (Karush-Kuhn-Tucker: KT)
  - Lets consider the first order Taylor series for the objective function and constraints around the optimum:

$$(1)\quad f(\boldsymbol{x}) \sim f(\boldsymbol{x}^*) + (\boldsymbol{x}-\boldsymbol{x}^*)^T \nabla f(\boldsymbol{x}^*) \qquad \left(\textit{for the objective function}\right)$$

$$(2)\quad h_i(\boldsymbol{x}) \sim h_i(\boldsymbol{x}^*) + (\boldsymbol{x}-\boldsymbol{x}^*)^T \nabla h_i(\boldsymbol{x}^*) = 0 \qquad \left(\textit{for the equality constraints } h_i(\boldsymbol{x}) = 0\right)$$

$$(3)\quad g_i(\boldsymbol{x}) \sim g_i(\boldsymbol{x}^*) + (\boldsymbol{x}-\boldsymbol{x}^*)^T \nabla g_i(\boldsymbol{x}^*) \le 0 \qquad \left(\textit{for the active inequality constraints } g_i(\boldsymbol{x}) \le 0 \textit{ with } g_i(\boldsymbol{x}^*) = 0\right)$$

– From eq. (1) we know that:

$$\nabla f(\boldsymbol{x}^*)^T(\boldsymbol{x}-\boldsymbol{x}^*) \geq 0$$

But now the allowed changes in ***x*** are those that satisfy the constraints.

– Since $h_i(\boldsymbol{x}^*)=0$, for ***x*** close enough to ***x****, we have from (2):

$$\nabla h_i(\boldsymbol{x}^*)^T(\boldsymbol{x}-\boldsymbol{x}^*)=0 \quad i=1,\cdots,p$$

– Since for the **active** inequality constraints, we have that $g_i(\boldsymbol{x}^*)=0$, from (3) we have:

$$\nabla g_i(\boldsymbol{x}^*)^T(\boldsymbol{x}-\boldsymbol{x}^*) \leq 0 \quad i = \textit{active}$$

- We can put together the restrictions from (2) and (3) as (where we have changed the sign and the inequality):

$$-\left(\sum_{i\in active} u_i \nabla g_i(\boldsymbol{x}^*) + \sum_{i=1}^{p} v_i \nabla h_i(\boldsymbol{x}^*)\right)^T (\boldsymbol{x}-\boldsymbol{x}^*) \geq 0$$

with $u_i \geq 0, v_i$ *arbitrary escalar*

- Putting this result in (1) we have that (1), (2) y (3) are satisfied if:

$$\nabla f(\boldsymbol{x}^*) = -\left(\sum_{i\in activas} u_i \nabla g_i(\boldsymbol{x}^*) + \sum_{i=1}^{p} v_i \nabla h_i(\boldsymbol{x}^*)\right)$$

- Hence, a necessary condition for $\boldsymbol{x}^*$ to be optimal is that there exist $u_i \geq 0$ and v_i such that:

$$\nabla f(\boldsymbol{x}^*) + \left(\sum_{i \in active} u_i \nabla g_i(\boldsymbol{x}^*) + \sum_{i=1}^{p} v_i \nabla h_i(\boldsymbol{x}^*) \right) = 0$$

  - Note that this means to put the **gradient of the objective function at the optimum** as a **linear combination of the constraint gradients. These must be linearly independent (regular).**
    - This condition is valid only for regular points. This does not mean that all optimum points must be regular.

- It is possible **to write the constrained optimization problem** using a function whose unconstrained optimization produces the same conditions that the constrained problem would produce.
  - Consider the function:

$$L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{s}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} u_i (g_i(\boldsymbol{x}) + s_i^2) + \sum_{i=1}^{p} v_i h_i(\boldsymbol{x})$$

    Known as Lagrangian function.
    - Note the new set of variables s_i

– The first order necessary conditions applied to this function:

$$\nabla L = \begin{pmatrix} \frac{\partial L}{\partial \boldsymbol{x}} \\ \frac{\partial L}{\partial \boldsymbol{u}} \\ \frac{\partial L}{\partial \boldsymbol{v}} \\ \frac{\partial L}{\partial \boldsymbol{s}} \end{pmatrix} = \begin{pmatrix} \nabla f(\boldsymbol{x}) + \left(\sum_{i=1}^{m} u_i \nabla g_i(\boldsymbol{x}) + \sum_{i=1}^{p} v_i \nabla h_i(\boldsymbol{x}) \right) \\ g_i(\boldsymbol{x}) + s_i^2 \quad i = 1, \cdots, m \\ h_i(\boldsymbol{x}) \quad i = 1, \cdots, p \\ 2 u_i s_i \quad i = 1, \cdots, m \end{pmatrix} = \boldsymbol{0}$$

- The first line is the known gradient condition but the sum of the equality constraints is done over all of them, not only over the active ones. The last line is satisfied only if either u_i or s_i are zero. If s_i=0, u_i can be ≠0 and viceversa. In this way, the u_i could be different from zero when the corresponding s_i=0. When s_i=0 the second line tells us that the constraint is active, hence the set of equations is equivalent to do the sum only over the active conditions. Since it is the fourth line the one that regulates this, it is know as the switching conditions. The third line is the equality constraints.

- Observation:
  - KT conditions, together with the switching conditions provide a ***set*** of systems of linear equations.
  - The set is specified by all possible combinations active/inactive of the constraints.
    - With 3 constraints we have 2^3 possible active/inactive combinations. Each combination produces a system of non-linear equations with a solution will produce, in general, a set of points that could be the optimum.
    - In general, we will not have cases with that many combinations, since the maximum number of equations will be limited by the dimension of the problem (there cannot be more active restrictions -that always include equality constraints- than variables)

– Summary: The KT conditions (candidate points)

For a point $\boldsymbol{x}^*$ to *be an optimum* $f(\boldsymbol{x})$ *constrained* to *conditions* $h_i(\boldsymbol{x})=0,\ \ i=1,\cdots,p \wedge g_i(\boldsymbol{x})\leq 0,\ \ i=1,\cdots,m$ *it must be satisfied*:

(1) $\boldsymbol{x}^*$ *must be a regular point.*

(2) $\nabla f(\boldsymbol{x}^*)+\left(\sum_{i=1}^{m} u_i \nabla g_i(\boldsymbol{x}^*)+\sum_{i=1}^{p} v_i \nabla h_i(\boldsymbol{x}^*)\right)=0$

(3) $\begin{Bmatrix} g_i(\boldsymbol{x}^*)+s_i^2=0 & i=1,\cdots,m \\ h_i(\boldsymbol{x}^*)=0 & i=1,\cdots,p \end{Bmatrix}$

(4) $u_i s_i=0 \quad i=1,\cdots,m$

(5) $s_i^2\geq 0 \quad i=1,\cdots,m$

(6) $u_i\geq 0 \quad i=1,\cdots,m$

The last two equations are only the condition of have to sum *a positive number* to *the inequality constraints* and *the condition of the* sum *of the constraints*

– Example: Minimize

$$f(x, y) = -x - y$$

$$\left\{ \begin{array}{l} g_1(x) = x + y^2 - 5 \leq 0 \\ g_2(x) = x - 2 \leq 0 \end{array} \right\}$$

The associated Lagrangian function:

$$L(x, y, u_{1,} u_{2,} s_{1,} s_2) = -x - y + (-5 + x + y^2 + s_1^2) u_1 + (x - 2 + s_2^2) u_2$$

The constraints – needed for the regularity condition:

$$\nabla g_1 = \begin{pmatrix} 1 \\ 2\mathrm{y} \end{pmatrix} \;\; ; \;\; \nabla g_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

The KT equations : $\nabla L = \mathbf{0}$

$$\begin{pmatrix} \dfrac{\partial L}{\partial x} \\ \dfrac{\partial L}{\partial y} \\ \dfrac{\partial L}{\partial u_1} \\ \dfrac{\partial L}{\partial u_2} \\ \dfrac{\partial L}{\partial s_1} \\ \dfrac{\partial L}{\partial s_2} \end{pmatrix} = \begin{pmatrix} u_1 + u_2 - 1 \\ 2 y u_1 - 1 \\ x + y^2 + s_1^2 - 5 \\ x + s_2^2 - 2 \\ 2 s_1 u_1 \\ 2 s_2 u_2 \end{pmatrix} = \mathbf{0}$$

Using the switching conditions:

Case (1): g_1 and g_2 *not active* $\rightarrow u_1 = u_2 = 0.$

The equations are:

$$\left\{\begin{array}{c} -1=0 \\ -1=0 \\ -5+x+y^2+s_1^2=0 \\ -2+x+s_2^2=0 \end{array}\right\}$$

That have no solution.

Case (2): g_1 *active* and g_2 *not active* $\rightarrow u_1 \neq 0 \quad (s_1 = 0), u_2 = 0$

The equations are:

$$\left\{\begin{array}{c} -1 + u_1 = 0 \\ -1 + 2\,y\,u_1 = 0 \\ -5 + x + y^2 = 0 \\ -2 + x + s_2^2 = 0 \end{array}\right\} \rightarrow \textit{Solution}: \left\{\begin{array}{c} x = 4.75 \\ y = 1/2 \\ u_1 = 1 \\ u_2 = 0 \\ s_1^2 = 0 \\ s_2^2 = -2.75 \end{array}\right\}$$

Since there is just one restriction the point is regular.

Case (3): g_2 *active* and g_1 *not active* $\rightarrow u_1=0 \ \ , u_2 \neq 0.$ $(s_2=0)$

The equations are:

$$\left\{ \begin{array}{c} -1+u_2=0 \\ -1=0 \\ -5+x+y^2+s_1^2=0 \\ -2+x=0 \end{array} \right\}$$

That have no solution.

Case (4): g_1 and g_2 *active* $\rightarrow u_1 \neq 0 \;\; (s_1 = 0), u_2 \neq 0. \;\; (s_2 = 0)$

The equations are:

$$\left\{\begin{array}{c} -1 + u_1 + u_2 = 0 \\ -1 + 2\, y\, u_1 = 0 \\ -5 + x + y^2 = 0 \\ -2 + x = 0 \end{array}\right\} \rightarrow \textit{Solution}\, 1: \left\{\begin{array}{c} x = 2 \\ y = -1.73 \\ u_1 = -0.289 \\ u_2 = 1.289 \\ s_1^2 = 0 \\ s_2^2 = 0 \end{array}\right\}$$

The gradients matrix *of the restrictions* in *that point*:

$$\left(\nabla g_1(x^*, y^*) \quad \nabla g_2(x^*, y^*)\right) = \begin{pmatrix} 1 & 1 \\ -3.46 & 0 \end{pmatrix} \rightarrow \textit{Rank } 2$$

The point is regular, although it could have been discarded beforehand since $u_1 < 0$

Case (4): *Solution 2*

$$\begin{Bmatrix} x=2 \\ y=1.73 \\ u_1=0.289 \\ u_2=0.711 \\ s_1^2=0 \\ s_2^2=0 \end{Bmatrix}$$

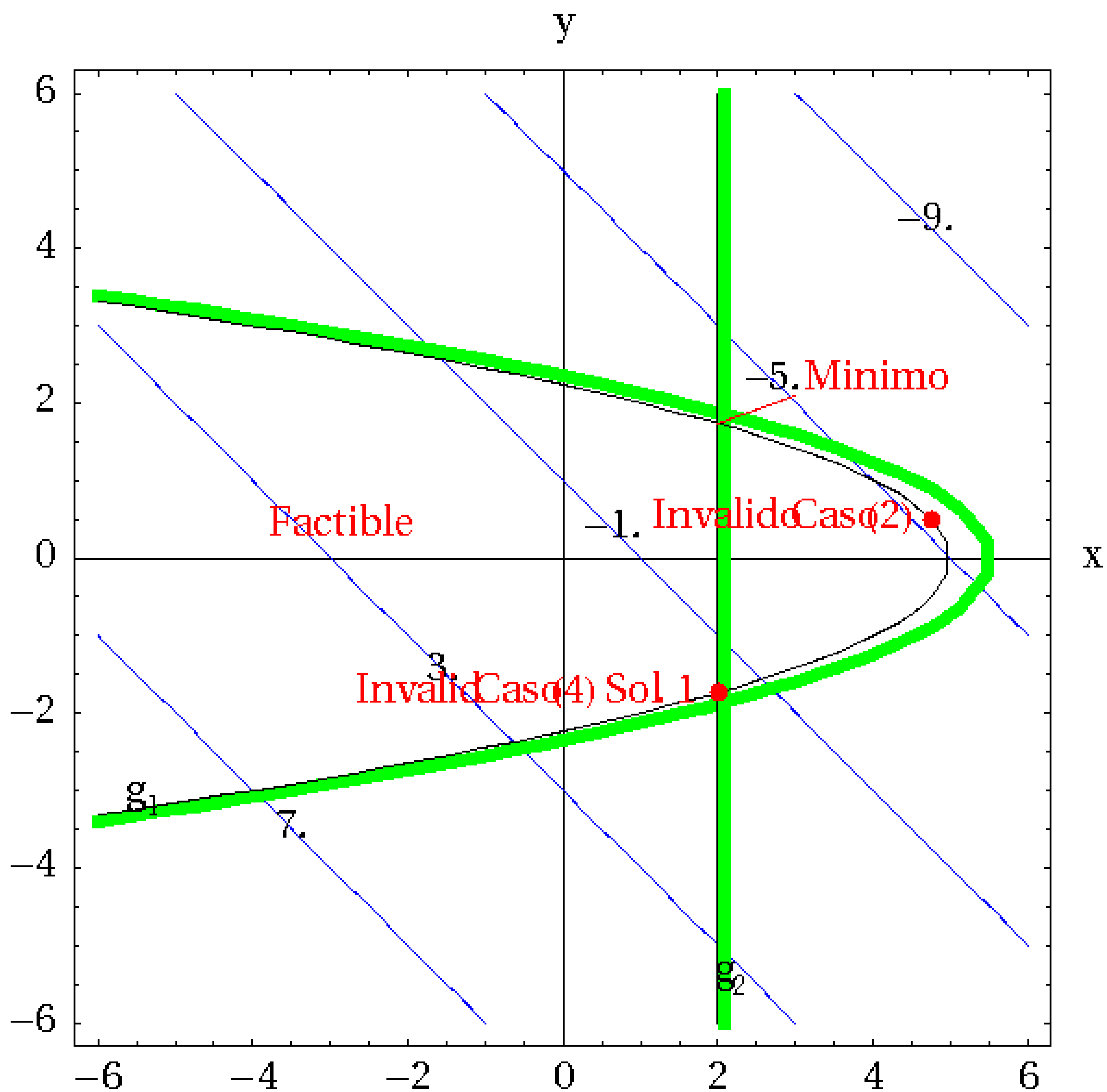
The matrix *of the constraint gradients* in *that point*:

$$\left(\nabla g_1(x^*, y^*) \quad \nabla g_2(x^*, y^*)\right) = \begin{pmatrix} 1 & 1 \\ 3.46 & 0 \end{pmatrix} \rightarrow \textit{Rank } 2$$

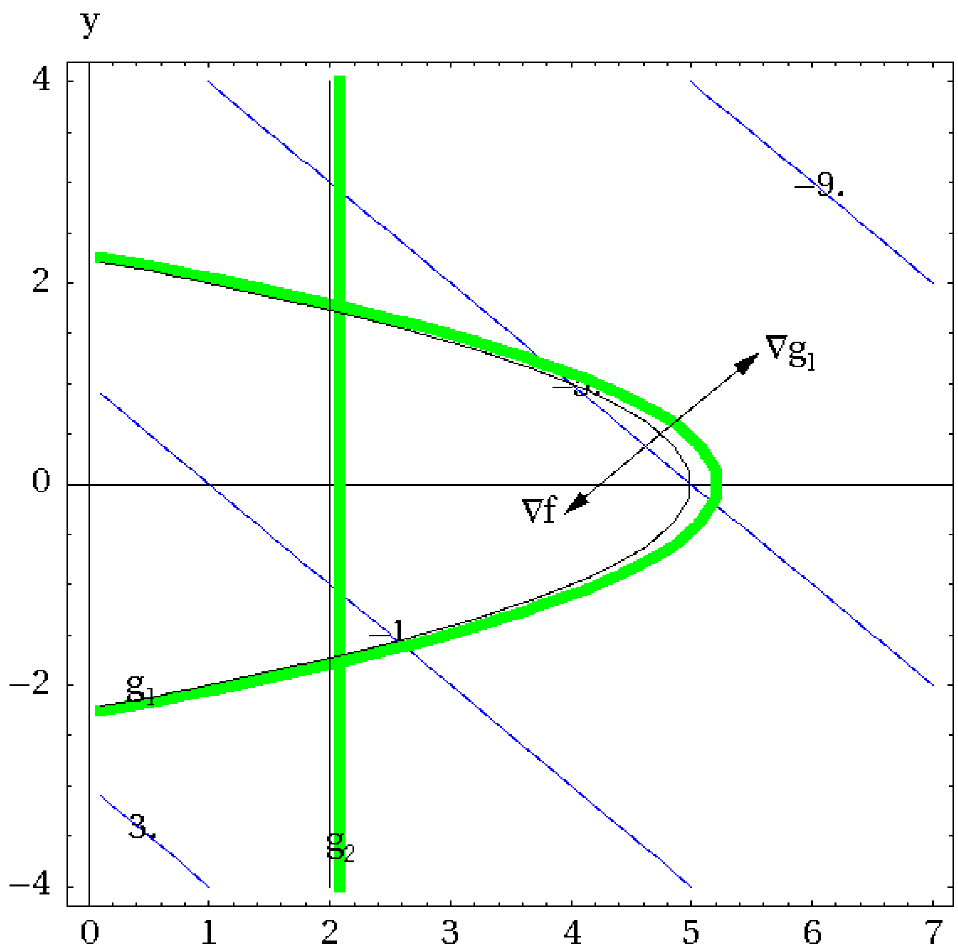
The point is regular, and *is a valid KT point.*

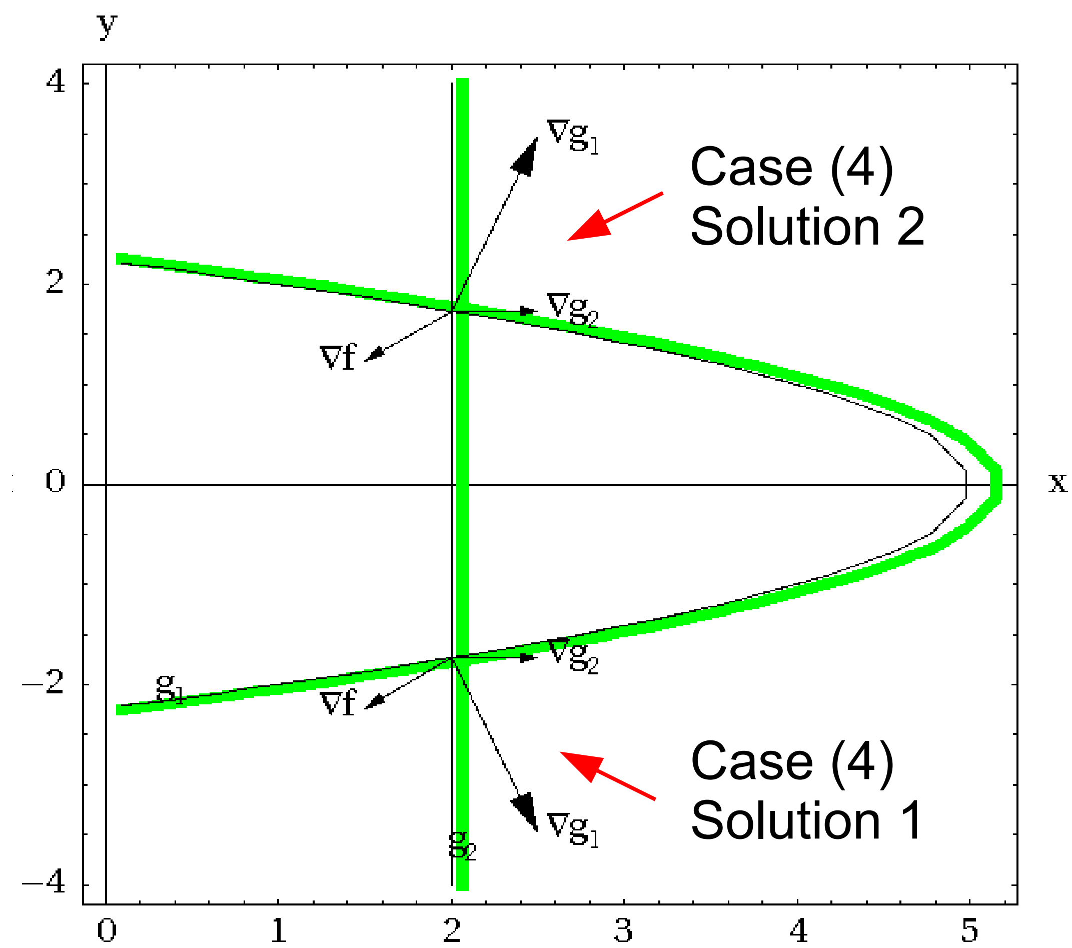
The value of the objective function is: -3.732

– The graphical representation of these cases:



- The meaning of the gradients:





In the point obtained in Case (2), the constraint g2 is not satisfied.

For the case (4) points: In the first solution there is no linear combination of the constraint's gradients with positive coefficients able to produce minus the gradient of the function. In the second Solution, there is.

- The case of convex problems.
  - If there is no nonlinear equality constraints and the inequality conditions and objective function are convex, then the problem is convex, hence the minimum is global.
  - In this case, the KT conditions are necessary and sufficient.

- Example:

$$f(x,y)=(x-2)^2+(y-3)^2$$
$$g(x,y)=(x-4)^2+(y-5)^2-6\leq 0$$

The function and *constraint Hessians* :

$$\begin{pmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial^2 g}{\partial x^2} & \dfrac{\partial^2 g}{\partial x \partial y} \\ \dfrac{\partial^2 g}{\partial y \partial x} & \dfrac{\partial^2 g}{\partial y^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix};$$

Minors: $A_1=2,\ \ A_2=4$ in *both cases* $\rightarrow$ f and g *convex.*
Since g *is an inequality* $\rightarrow$ *The problem is convex.*

The associated Lagrangian function:

$$L(x, y, u_{1,} s_1)=(x-2)^2+(y-3)^2+u_1\left((x-4)^2+(y-5)^2-6+s_1^2\right)$$

The complete KT equations: $\nabla L=\mathbf{0}$

$$\begin{pmatrix} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial y} \\ \frac{\partial L}{\partial u_1} \\ \frac{\partial L}{\partial s_1} \end{pmatrix}=\begin{pmatrix} -4+2x-8u_1+2xu_1 \\ -6+2y-10u_1+2yu_1 \\ -6+(-4+x)^2+(y-5)^2+s_1^2 \\ 2s_1u_1 \end{pmatrix}=\mathbf{0}$$

With unique solution: $\begin{pmatrix} x^*=2.267 \\ y^*=3.267 \\ u_1=0.154 \end{pmatrix}$; $s_1=0$ *since there is only one constraint, it is also regular, hence it is a valid solution.*

- Non convex example:

$$f(x,y)=(x-2)^2+(y-3)^2$$
$$g(x,y)=(x-4)^2+(y-5)^2-6=0$$

Since there is a nonlinear equality constraint the problem is not convex. The associated Lagrangian function:

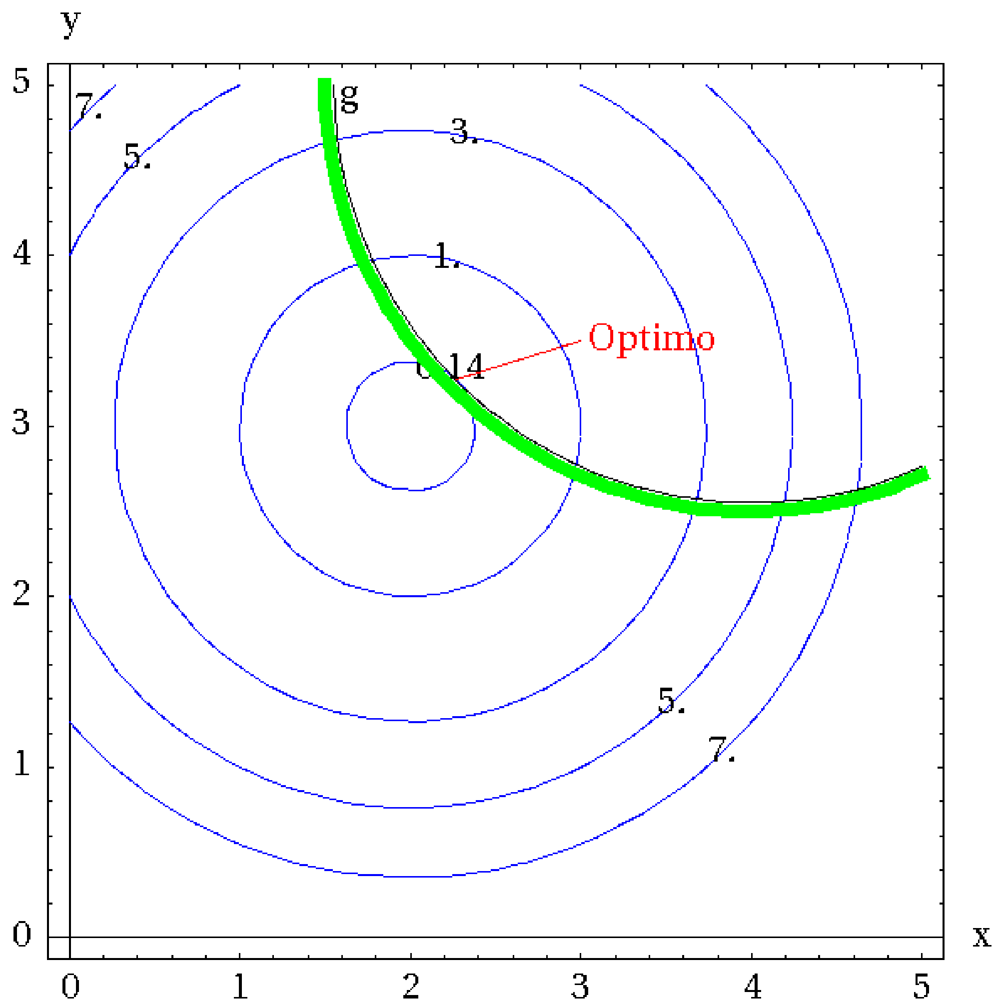
$$L(x,y,v_1)=(x-2)^2+(y-3)^2+v_1\left((x-4)^2+(y-5)^2-6\right)$$

The KT equations:

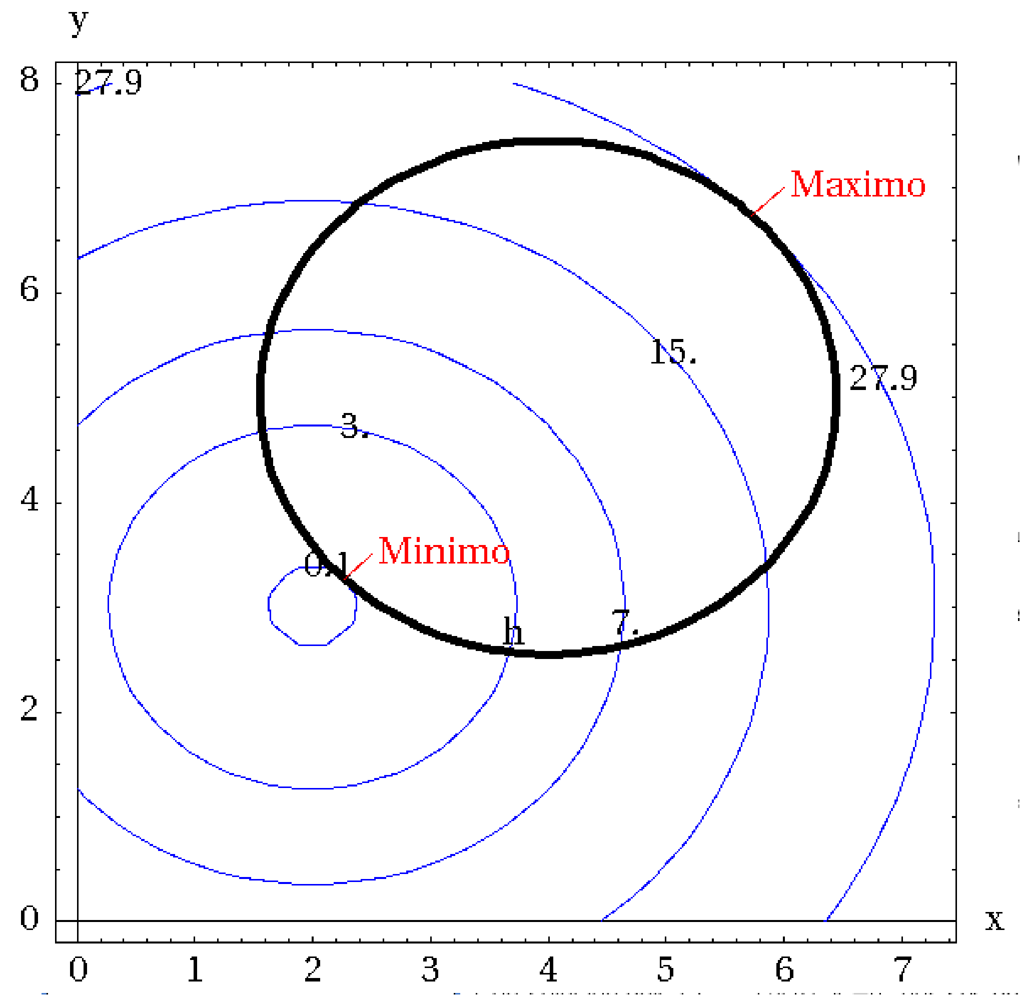
$$\begin{pmatrix} \dfrac{\partial L}{\partial x} \\ \dfrac{\partial L}{\partial y} \\ \dfrac{\partial L}{\partial v_1} \end{pmatrix} = \begin{pmatrix} -4+2x-8v_1+2xv_1 \\ -6+2y-10v_1+2yv_1 \\ -6+(-4+x)^2+(y-5)^2 \end{pmatrix} = \mathbf{0}$$

With valid solutions: $Sol_1=\begin{pmatrix} x^*=2.267 \\ y^*=3.267 \\ v_1=0.154 \end{pmatrix}$ and $Sol_2=\begin{pmatrix} x^*=5.732 \\ y^*=6.732 \\ v_1=-2.154 \end{pmatrix}$.

- Graphically:



Convex case



Non Convex Case

- Second order sufficient conditions:
  - KT conditions, except in the convex case, are only necessary.
  - Using the Lagrangian as an unconstrained minimization problem, the sufficient second order conditions in a KT point $\{x^*, u^*, v^*\}$:

$$\boldsymbol{d}^T\left(\nabla^2 f(\boldsymbol{x}^*) + \sum_{i\in active} u_i^* \nabla^2 g_i(\boldsymbol{x}^*) + \sum_{i=1}^{p} v_i^* \nabla^2 h_i(\boldsymbol{x}^*)\right)\boldsymbol{d} > 0$$

$$\nabla g_i(\boldsymbol{x}^*)^T \boldsymbol{d} = 0 \quad i = \mathit{active}; \qquad \nabla h_i(\boldsymbol{x}^*)^T \boldsymbol{d} = 0 \quad i = 1, \cdots, p$$

    - In the unconstrained case this means that we need a positive definite Hessian. In the constrained case, only those displacements ***d*** that are compatible with the constraints are allowed (second line of equations).

- In the general case, the feasible changes have to be determined through the second line of equations.

- If

$$\nabla^2 f(\boldsymbol{x}^*) + \sum_{i \in active} u_i^* \nabla^2 g_i(\boldsymbol{x}^*) + \sum_{i=1}^{p} v_i^* \nabla^2 h_i(\boldsymbol{x}^*)$$

  Is positive definite, then the condition is valid for all ***d*** and there is no need to calculate them.

- Example: $$f(x,y) = -x^2 + y$$ $$h(x) = -x^2 - y^2 + 1 = 0$$

The associated Lagrangian function will be:

$$L(x, y, v_1) = -x^2 + y + (1 - x^2 - y^2) v_1$$

The gradients of the constraints will not be needed to check *regularity since there is just one.*

The full KT equations: $\nabla L = \mathbf{0}$

$$\begin{pmatrix} \dfrac{\partial L}{\partial x} \\ \dfrac{\partial L}{\partial y} \\ \dfrac{\partial L}{\partial v_1} \end{pmatrix} = \begin{pmatrix} -2x - 2x v_1 \\ 1 - 2y v_1 \\ 1 - x^2 - y^2 \end{pmatrix} = \mathbf{0}$$

Have four solutions:

$$s_1=\begin{Bmatrix} f=-1.25 \\ x=-0.866 \\ y=-0.5 \\ v_1=-1 \end{Bmatrix} \quad s_2=\begin{Bmatrix} f=-1 \\ x=0 \\ y=-1 \\ v_1=-0.5 \end{Bmatrix} \quad s_3=\begin{Bmatrix} f=1 \\ x=0 \\ y=-1 \\ v_1=0.5 \end{Bmatrix} \quad s_4=\begin{Bmatrix} f=-1.25 \\ x=0.866 \\ y=-0.5 \\ v_1=-1 \end{Bmatrix}$$

Since there is just one constraint the points are regular.

The second order conditions are then reduced to:

$$\boldsymbol{d}^T\left(\nabla^2 f(\boldsymbol{x}^*)+v_1^*\nabla^2 h(\boldsymbol{x}^*)\right)\boldsymbol{d}>0$$

$$\nabla h(\boldsymbol{x}^*)^T\boldsymbol{d}=0$$

The Hessian matrices are:

$$\nabla^2 f=\begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} \;;\; \nabla^2 h=\begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}$$

- Second order conditions in *s1:*

$$\nabla h = \begin{pmatrix} -2x \\ -2y \end{pmatrix} = \begin{pmatrix} 1.73 \\ 1 \end{pmatrix}$$

$$\nabla h\, \boldsymbol{d} = \begin{pmatrix} 1.73 \\ 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = 1.73\, d_1 + d_2 = 0 \rightarrow \boldsymbol{d}^T = (-0.57\, d_{2,}\, d_2)$$

Substituting in the second order condition:

$$(-0.57\, d_{2,}\, d_2) \left[\begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} + (-1) \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} \right] \begin{pmatrix} -0.57\, d_2 \\ d_2 \end{pmatrix} = d_2^2$$

Which is always >0, hence the condition is fulfilled and it is a local minimum.

- Second order conditions in *s2:*

$$\nabla h = \begin{pmatrix} -2x \\ -2y \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\nabla h\, \boldsymbol{d} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = 2d_2 = 0 \rightarrow \boldsymbol{d}^T = (d_{1,} 0)$$

Substituting in the second order condition:

$$(d_{1,} 0) \left[\begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} + (-0.5) \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} \right] \begin{pmatrix} d_1 \\ 0 \end{pmatrix} = -d_1^2$$

Which is always < 0, hence the condition is not fulfilled and It is NOT a local minimum.

- Second order conditions in *s3:*

$$\nabla h = \begin{pmatrix} -2x \\ -2y \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

$$\nabla h\,\boldsymbol{d} = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -2d_2 = 0 \rightarrow \boldsymbol{d}^T = (d_{1,} 0)$$

Substituting in the second order condition:

$$(d_{1,} 0) \left[\begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} + 0.5 \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} \right] \begin{pmatrix} d_1 \\ 0 \end{pmatrix} = -3d_1^2$$

Which is always < 0, hence the condition is not fulfilled and
It is NOT a local minimum.

- Second order conditions in *s4:*

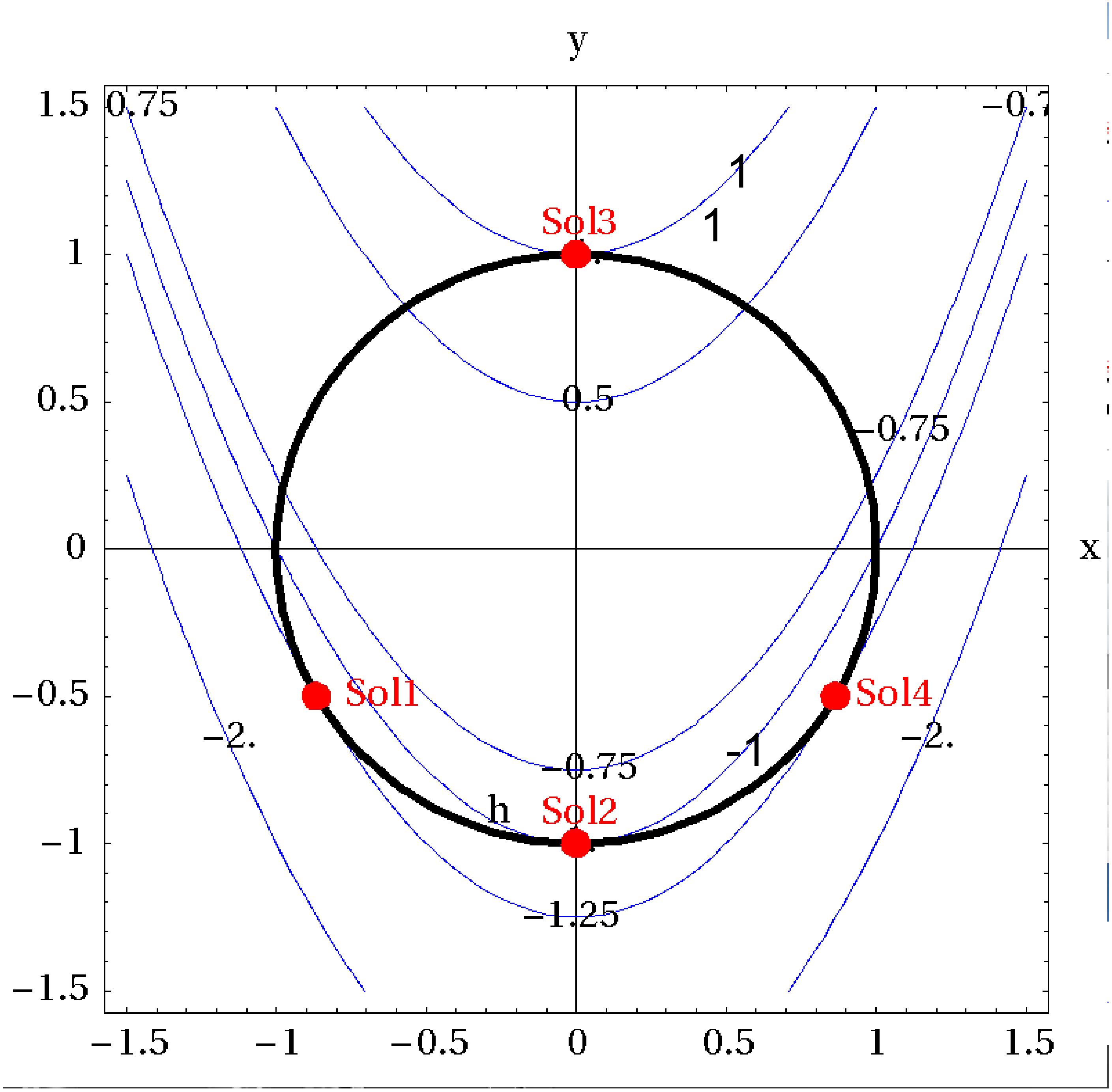
$$\nabla h = \begin{pmatrix} -2x \\ -2y \end{pmatrix} = \begin{pmatrix} -1.73 \\ 1 \end{pmatrix}$$

$$\nabla h \boldsymbol{d} = \begin{pmatrix} -1.73 \\ 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -1.73 d_1 + d_2 = 0 \rightarrow \boldsymbol{d}^T = (0.577 \mathrm{d}_{2,} d_2)$$

Substituting in the second order condition:

$$(0.577 d_{2,} d_2) \left[\begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} + (-1) \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} \right] \begin{pmatrix} 0.577 d_2 \\ d_2 \end{pmatrix} = 2 d_2^2$$

Which is always > 0, hence the condition is fulfilled and
We have a local minima.


y
x
Sol1
Sol2
Sol3
Sol4
h
0.75
-0.7
1
1
0.5
-0.75
-0.75
-1
-1.25
-2.
-2.
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

Unconstrained Minimization

- Find the vector ***x*** minimizing ***f(x).***
  - Iterative method: build a sucession starting in a given point: $\boldsymbol{x}^{\mathbf{0}}$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} + \alpha_k \boldsymbol{d}^{k} \quad k = 0, 1, \cdots$$

- The directions ***d*** are the descent directions and are calculated at each step. α is a scalar: the size of the step given in direction ***d***
- At each step α and ***d*** are selected such as f($\boldsymbol{x}^{k+1}$) < f($\boldsymbol{x}^{k}$)
- The stopping conditions are varied, a typical one is:

$$\|\nabla f(\boldsymbol{x}^{k+1})\| = \sqrt{\left(\frac{\partial(\boldsymbol{x}^{k+1})}{\partial x_1}\right)^2 + \cdots + \left(\frac{\partial(\boldsymbol{x}^{k+1})}{\partial x_n}\right)^2} \le tol \quad \textit{typical} \sim 10^{-3}$$

- Steepest descent direction:
  - To calculate ***d*** such that f($\boldsymbol{x}^{k+1}$) < f($\boldsymbol{x}^{k}$) we use a first order Taylor series:

$$f\left(\boldsymbol{x}^{k+1}\right)=f\left(\boldsymbol{x}^{k}+\alpha_{k}\boldsymbol{d}^{k}\right)$$

$$f\left(\boldsymbol{x}^{k}\right)+\alpha_{k}\nabla f\left(\boldsymbol{x}^{k}\right)^{T}\boldsymbol{d}^{k}<f\left(\boldsymbol{x}^{k}\right)$$

$$\nabla f\left(\boldsymbol{x}^{k}\right)^{T}\boldsymbol{d}^{k}<0$$

    - Where only positive steps (α) are considered. The bigger the absolute value of $\nabla f\left(\boldsymbol{x}^{k}\right)^{T}\boldsymbol{d}^{k}$, the bigger the descent speed in that direction ***d***

– Example: Determine if (1,1), (-1,1) y (31,12) are descnt diretions from the point (1,2) and with respect to the function:

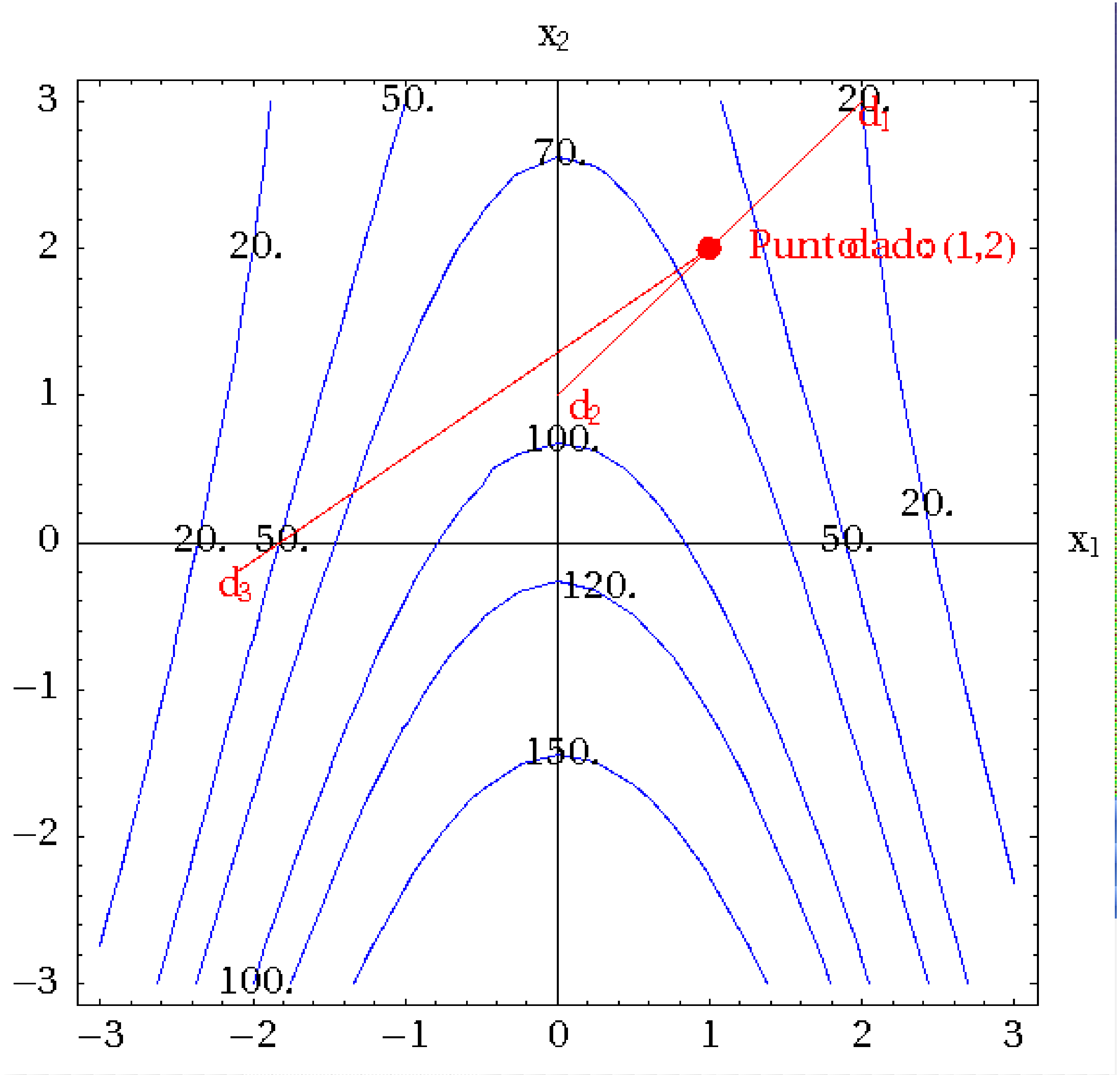
$$f(x_{1,} x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)$$

Which is the steepest descent direction?

- The gradient is:

$$\nabla f = \begin{pmatrix} 1 - 44x_1 + 4x_1^3 + 4x_1x_2 \\ -22 + 2x_1^2 + 4x_2 \end{pmatrix} \; \textit{evaluated} \text{ in } (1,2) \;\; \nabla f(1,2) = \begin{pmatrix} -31 \\ -12 \end{pmatrix}$$

- Con ***d***= (1,1): $\begin{pmatrix} -31 \\ -12 \end{pmatrix}(1,1) = -43 < 0 \rightarrow \textit{descent}$
- Con ***d***= (-1,1): $\begin{pmatrix} -31 \\ -12 \end{pmatrix}(-1,1) = 19 > 0 \rightarrow \textit{non descent}$
- Con ***d***= (31,12): $\begin{pmatrix} -31 \\ -12 \end{pmatrix}(31,12) = -1105 < 0 \rightarrow$ $\begin{matrix} \textit{steepest descent}: \\ d = -\nabla f(1,2) \end{matrix}$



x_2
x_1
50.
20.
70.
d_1
Puntodado (1,2)
d_2
100.
120.
150.
d_3

- Step size: once selected the direction ***d***, α is calculated to obtain the minimum of ***f*** along ***d.***
  - If there is an analytical expression $\phi(\alpha)=f(\boldsymbol{x}^k+\alpha_k \boldsymbol{d}^k)$ it is possible to try a direct minimization, leading to:

$$\frac{d\phi}{d\alpha}=\frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial \alpha}+\frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial \alpha}+\cdots=\nabla f(\boldsymbol{x}^{k+1})^T \boldsymbol{d}^k=0$$

    where the chain rule has been used treating *f* as a function of α and $\boldsymbol{x}=\boldsymbol{x}^k+\alpha \boldsymbol{d}^k$ *hence* $\partial \boldsymbol{x}/\partial \alpha=\boldsymbol{d}^k$
  - Usually this will not be the case, hence a numerical minimization along a line is used.

- Numerical line minimization:
  - Equal interval search:
    - We start with a lower bound for α, ej: $\alpha=0$. Its value is then increased with a value δ, ej: $\delta=0.5$

      (1) Let $\alpha 1=\alpha$

      (2) Let $\alpha 2= \alpha 1+\delta$

      (3) If $\Phi(\alpha 2) <= \Phi(\alpha 1)$, the minimum is in between $\alpha 1$ and $\alpha 2$ or beyond $\alpha 2$. Then let $\alpha 1= \alpha 2$ and go to step (2).

      (4) If $\Phi(\alpha 2) > \Phi(\alpha 1)$, the minima has been overshooted, it is in between $\alpha 1-\delta$ and $\alpha 2$. δ is then lowered (i.e.: divide by 10) and repeat.
    - The algorithm ends when $\alpha 2-(\alpha 1-\delta) < tol$. In this case $\alpha min=(\alpha 2+(\alpha 1-\delta))/2$ is used.

- Golden search method.
  - Similar to the previous one, but the step size is increased with the number of the iteration:

$$\alpha_2=\alpha_1+\tau^{n-1}\delta \quad \textit{with} \quad \tau=\frac{1+\sqrt{5}}{2} \quad \text{and} \quad n= \textit{ iteration number}$$

    - The search for bounds is faster because the step size is increased with each iteration since τ>1~1.62
  - A refinement phase is included: the improvement of the upper, αu, and lower bound, αl, is done using two new points between these: αa y αb. In these points it is satisfied:
    - If Φ(αa) < Φ(αb), then the minimum is in the interval (αa,αb)
    - If Φ(αa) >= Φ(αb), then the minimum is in the interval (αa,αu)

- Algorithm:
  - Start with a lower bound for α, ej: α=0. Increase it with a given δ value, eg: δ=0.5

    (1) Let α1=α, n=1

    (2) Let α2= α1+τ^{n-1} δ

    (3) If Φ(α2) <= Φ(α1), the minimum is in between α1 and α2 or beyond α2. Let α1= α2 and go to step (2).

    (4) If Φ(α2) > Φ(α1), we've gone beyond the minimum, hence it is in between α1-δ and α2. A refinement phase is done with values:

    $$\alpha_a = \alpha_l + \left(1 - \frac{1}{\tau}\right)(\alpha_u - \alpha_l) \quad y \quad \alpha_b = \alpha_l + \frac{(\alpha_u - \alpha_l)}{\tau}$$

    With these values, either αa or αb are the same in the next step, hence calculations are reused.
  - The algorithm finishes when α2-(α1-δ) < tol. As the minimum value is used αmin=(α2+(α1-δ))/2

- Cuadratic Interpolation:
  - The previous methods do not consider the shape of the function. If we have three points αl, αm and αu its interpolation parabola can be calculated and its minimum will be given by:

$$\alpha_q = \frac{1}{2}\left(\frac{\phi_l(\alpha_m^2-\alpha_u^2)+\phi_m(\alpha_u^2-\alpha_l^2)+\phi_u(\alpha_l^2-\alpha_m^2)}{\phi_l(\alpha_m-\alpha_u)+\phi_m(\alpha_u-\alpha_l)+\phi_u(\alpha_l-\alpha_m)}\right)$$

- The algorithm is then:

  (1)If αq <= αm

  - If Φ(αm) >= Φ(αq) then the minimum is in (αl,αm), hence (αl,αq,αm) are used for the next interpolation round.
  - If Φ(αm) < Φ(αq) then the minimum is in (αq,αu), then (αq,αm,αu) are used.

  (2)If αq > αm

  - If Φ(αm) >= Φ(αq) then the minimum is in (αm,αu) and (αm,αq,αu) are used.
  - If Φ(αm) < Φ(αq) then the minimum is in (αl,αq) and (αl,αm,αq) are used.

- A usual termination criteria:

$$\left|\frac{\phi(\alpha_q)-\phi(\alpha_q)}{\phi(\alpha_q)}\right| \leq tol$$

- Approximated Search:
  - Line minimization is used so often that it is very convenient to have a fast search, however only approximate.
  - Armijo's Rule:
    - Use as the α of the minimum one that satisfies:

$$\phi(\alpha) \leq \phi(0) + \alpha \epsilon \phi'(0) \quad \text{and} \quad \phi(\eta \alpha) \leq \phi(0) + \alpha \eta \epsilon \phi'(0)$$

$0<\varepsilon<1$ y $\eta>1$ are user specified parameters (Eg.: $\varepsilon = 0.2$ y $\eta>2$)

– Algorithm:

(1) Start with an arbitrary α value.

(2) Calculate $\Phi(\alpha)$.

(a) If $\Phi(\alpha) <= \Phi(0) + \alpha\varepsilon\, \Phi'(0)$, increase α using $\eta\alpha$ and repeat till failure. Use as the step length the last α that passed the test.

(b) If $\Phi(\alpha) > \Phi(0) + \alpha\varepsilon\, \Phi'(0)$ (with the initial α value), reduce α using α/η and test (a) till some α passes the test.

– Example of line minimization. Function:

$$\phi(\alpha) = 1 - \frac{1}{1 - \alpha + 2\alpha^2}$$

1) Equal interval searches: α=0, δ=0.1, interval dividing factor=5,tol=0.01
(12 φ(α) evaluations)

$\delta \to 0.1$

α	$\phi(\alpha)$
0	0
0.1	-0.0869565
0.2	-0.136364
0.3	-0.136364
0.4	-0.0869565

Bounds → [0.2, 0.4]

$\delta \to 0.02$

α	$\phi(\alpha)$
0.2	-0.136364
0.22	-0.140511
0.24	-0.142596
0.26	-0.142596
0.28	-0.140511

Bounds → [0.24, 0.28]

$\delta \to 0.004$

α	$\phi(\alpha)$
0.24	-0.142596
0.244	-0.142763
0.248	-0.142847
0.252	-0.142847
0.256	-0.142763

Bounds → [0.248, 0.256]

- {0.252, {0.248, 0.256}}

2) Golden search: 13 evaluations α=0, δ=0.1, interval divider=5, tol=0.01

```
******** Bounding phase ********

δ → 0.1
```

α	ϕ(α)
0	0
0.1	-0.0869565
0.261803	-0.142493
0.523607	0.024125

Bounds → {0.1, 0.523607}

```
******** Refinement phase ********
```

α_L	α_U	α_a	α_b	ϕ(αa)	ϕ(αb)	I
0.1	0.523607	0.261803	0.361803	-0.142493	-0.111111	0.423607
0.1	0.361803	0.2	0.261803	-0.136364	-0.142493	0.261803
0.2	0.361803	0.261803	0.3	-0.142493	-0.136364	0.161803
0.2	0.3	0.238197	0.261803	-0.142493	-0.142493	0.1
0.238197	0.3	0.261803	0.276393	-0.142493	-0.14104	0.0618034
0.238197	0.276393	0.252786	0.261803	-0.142837	-0.142493	0.0381966
0.238197	0.261803	0.247214	0.252786	-0.142837	-0.142837	0.0236068
0.247214	0.261803	0.252786	0.256231	-0.142837	-0.142756	0.0145898
0.247214	0.256231	0.250658	0.252786	-0.142856	-0.142837	0.00901699

=

0.251722

3) Approximated search using Armijo's rule: initial α =1, η=2, ε=0.2. 3 evaluations.

$\phi(0) \to 0 \quad \phi'(0) \to -1$

α	$\phi(\alpha)$	$\phi(0) + \alpha\phi'(0)\epsilon$	$\phi(0) + \alpha\phi'(0)\eta\epsilon$
1.	0.5	-0.2	--
0.5	0.	--	-0.2
0.25	-0.142857	--	-0.1

0.25

- Unconstrained Minimization: Steepest descent.
  - Choose as a descent direction the steepest: Minus the gradient.

$$\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$$

    - The method will produce directions perpendicular to the previous one since, in the line minimization performed at each step:

$$\frac{d\phi}{d\alpha} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial \alpha} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial \alpha} + \cdots = \nabla f(\boldsymbol{x}^{k+1})^T \boldsymbol{d}^k = 0$$

    - This makes for a fast approach when far away from the minimum, but approaching in a zig-zag fashion and short steps, as happens close to the minimum is slow.

– Example: Minimization using the steepest descent applied to the function: $f(x,y)=(x+y)^2+\left(2(x^2+y^2-1)-\frac{1}{3}\right)^2$

$$f \to (x+y)^2 + \left(-\frac{1}{3} + 2\,(-1 + x^2 + y^2)\right)^2$$

$$\nabla f \to \begin{pmatrix} -\frac{50\,x}{3} + 16\,x^3 + 2\,y - 16\,x\,y^2 \\ 2\,x - \frac{50\,y}{3} + 16\,x^2\,y + 16\,y^3 \end{pmatrix}$$

***** Iteration 1 ***** Current point → {-1.25, 0.25}

Direction-finding phase:

$$\nabla f(x) \to \begin{pmatrix} -11.1667 \\ -0.166667 \end{pmatrix} \quad d \to \begin{pmatrix} 11.1667 \\ 0.166667 \end{pmatrix}$$

Only the first step is shown.
Analytical minimization is used.

$$||\nabla f(x)|| \to 11.1679 \quad f(x) \to 1.84028$$

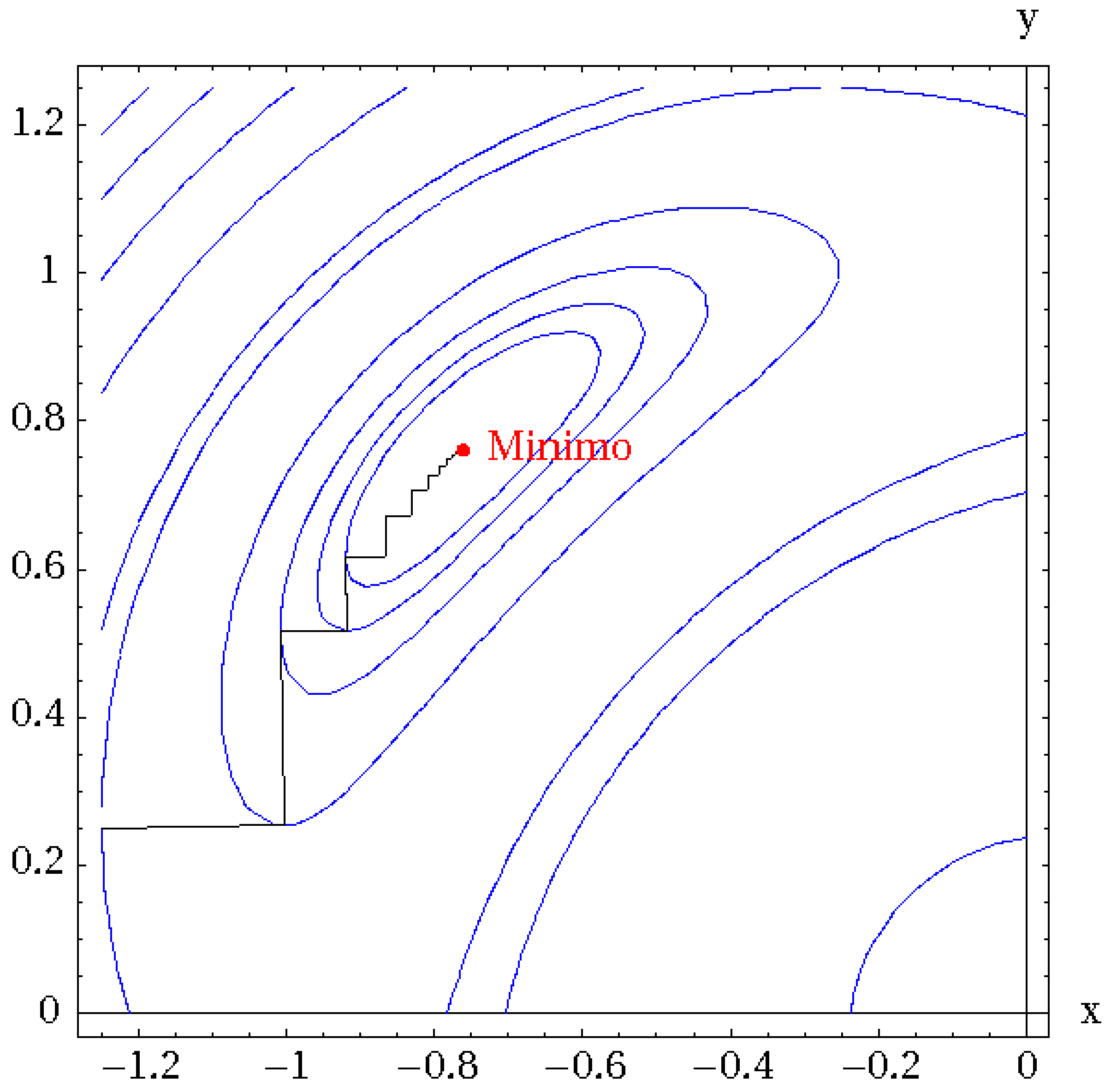
Step-length calculation phase:

$$\mathrm{xk1} \to \begin{pmatrix} -1.25 + 11.1667\,\alpha \\ 0.25 + 0.166667\,\alpha \end{pmatrix}$$

$$\nabla f(\mathrm{xk1}) \to \begin{pmatrix} 22283.7\,(-0.198059+\alpha)\,(-0.115053-\alpha)\,(-0.0219909+\alpha) \\ 332.593\,(-0.182098+\alpha)\,(0.00188867+\alpha)\,(1.45705+\alpha) \end{pmatrix}$$

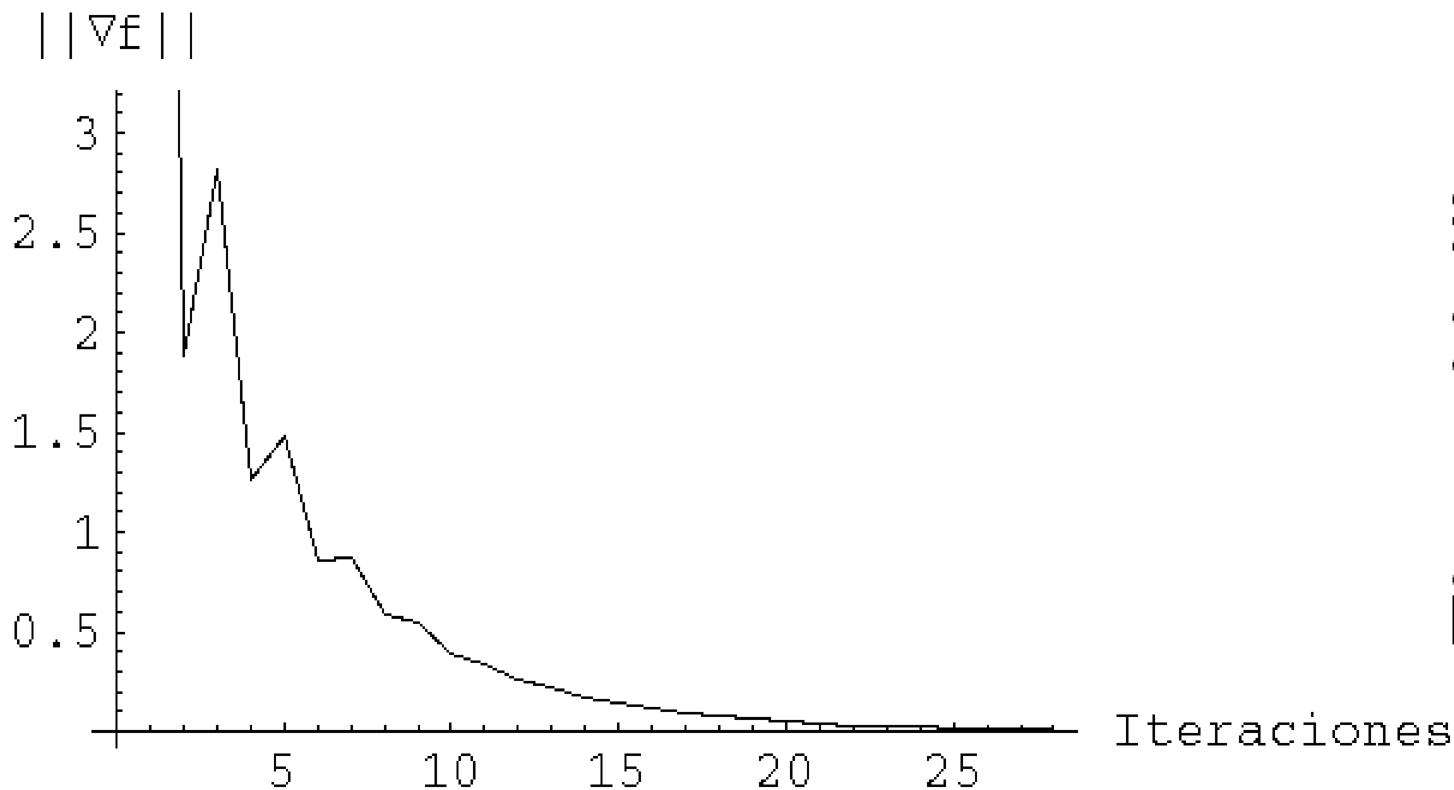
$$d\phi/d\alpha \equiv \nabla f(\mathrm{xk1}).d=0 \to 248890.\,(-0.197979+\alpha)\,(-0.114697+\alpha)\,(-0.0220681+\alpha) == 0$$

$$\alpha \to 0.0220681$$



Note the zig-zag pattern in approaching the minimum..

Using the norm of the gradient, that must be zero in a minimum, as a convergence indicator.



- Conjugate gradients: To improve the convergence by avoiding the zig-zag approach, conjugate directions are used.
  - The steepest descent direction is modified mixing it with the direction in the previous step:

$$\boldsymbol{d}^k = -\nabla f(\boldsymbol{x})^k + \beta \boldsymbol{d}^{k-1}$$

- Two typical formulas for β are used:

$$\textit{Fletcher-Reeves:}\quad \beta = \frac{\left(\nabla f(\boldsymbol{x}^k)\right)^T \nabla f(\boldsymbol{x}^k)}{\left(\nabla f(\boldsymbol{x}^{k-1})\right)^T \nabla f(\boldsymbol{x}^{k-1})}$$

$$\textit{Polak-Ribiere:}\quad \beta = \frac{\left(\nabla f(\boldsymbol{x}^{k-1}) - \nabla f(\boldsymbol{x}^k)\right)^T \nabla f(\boldsymbol{x}^k)}{\left(\nabla f(\boldsymbol{x}^{k-1})\right)^T \nabla f(\boldsymbol{x}^{k-1})}$$

  - As a general rule, the β value, must modify the steepest descent direction only slightly when far away from the minimum and more when approaching it.

- Example: Conjugate Gradients, same function.

$$f \to (x+y)^2 + \left(-\frac{1}{3} + 2\left(-1 + x^2 + y^2\right)\right)^2$$

$$\nabla f \to \begin{pmatrix} -\frac{50x}{3} + 16x^3 + 2y + 16xy^2 \\ 2x - \frac{50y}{3} + 16x^2 y + 16y^3 \end{pmatrix}$$

Using the PolakRibiere method with Analytical line search

***** Iteration 1 ***** Current point $\to$ [-1.25, 0.25]

Direction-finding phase:

$$\nabla f(x) \to \begin{pmatrix} -11.1667 \\ -0.166667 \end{pmatrix} \quad d \to \begin{pmatrix} 11.1667 \\ 0.166667 \end{pmatrix}$$

$||\nabla f(x)|| \to 11.1679 \quad \beta \to 0.$

The first step is the same than the Steepest descent: β=0

$f(x) \to 1.84028$

Step-length calculation phase:

$$xk1 \to \begin{pmatrix} -1.25 + 11.1667\alpha \\ 0.25 + 0.166667\alpha \end{pmatrix}$$

$$\nabla f(xk1) \to \begin{pmatrix} 22283.7(-0.198059+\alpha)(-0.115053+\alpha)(-0.0219909+\alpha) \\ 332.593(-0.182098+\alpha)(0.00188867+\alpha)(1.45705+\alpha) \end{pmatrix}$$

$d\phi/d\alpha = \nabla f(xk1).d=0 \to 248890.(-0.197979+\alpha)(-0.114697+\alpha)(-0.0220681+\alpha) == 0$

$\alpha \to 0.0220681$

***** Iteration 2 ***** Current point → [-1.00357, 0.253678]

Direction-finding phase:

$$\nabla f(x) \to \begin{pmatrix} 0.0281495 \\ -1.88601 \end{pmatrix} \quad d \to \begin{pmatrix} 0.290392 \\ 1.89077 \end{pmatrix}$$

$\|\nabla f(x)\| \to 1.88622 \quad \beta \to 0.0285261$

$f(x) \to 0.598561$

Step-length calculation phase:

$$xk1 \to \begin{pmatrix} -1.00357 + 0.290392\alpha \\ 0.253678 + 1.89077\alpha \end{pmatrix}$$

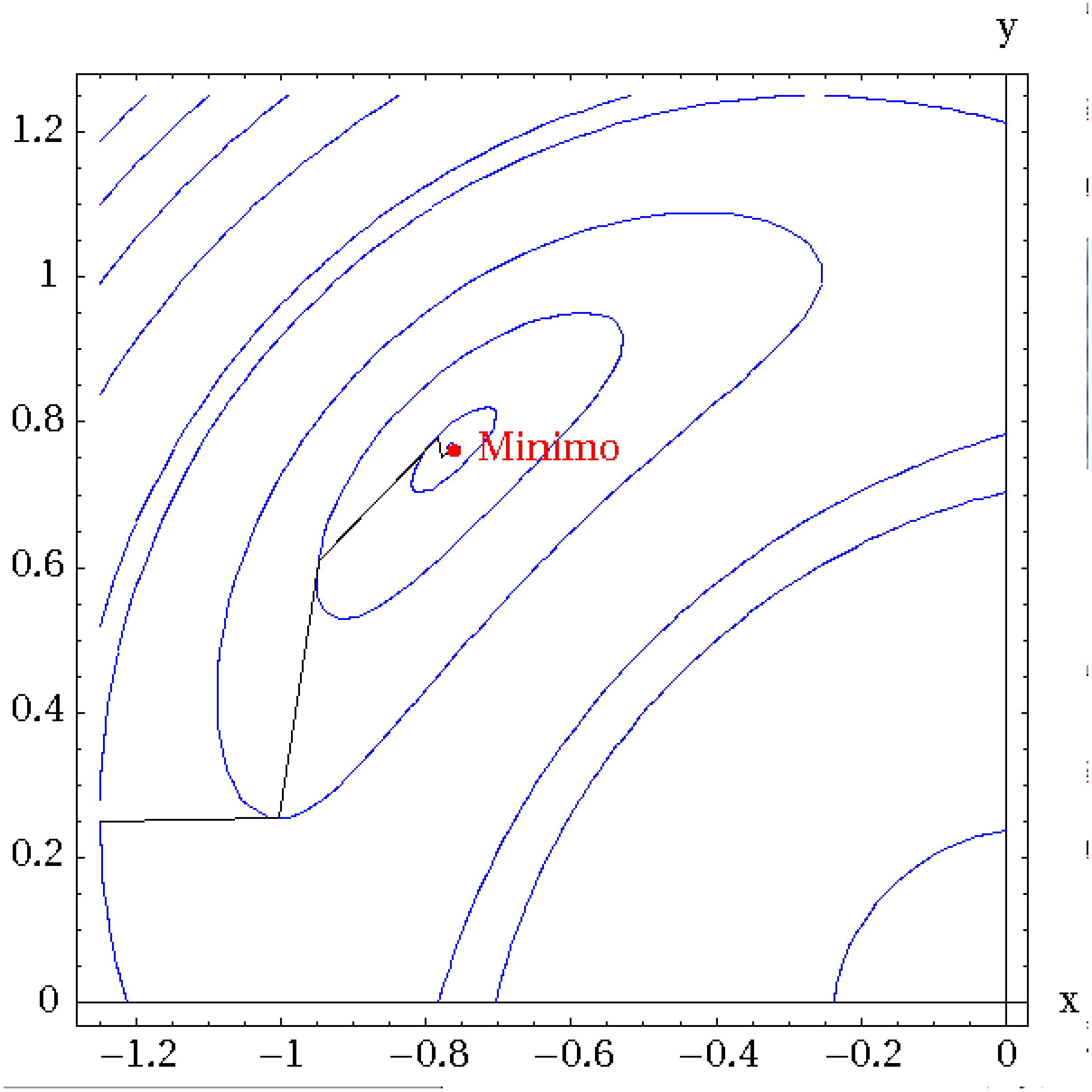
$$\nabla f(xk1) \to \begin{pmatrix} 17.0023\,(-3.38977+\alpha)\,(-0.0103728+\alpha)\,(0.0470866+\alpha) \\ 110.703\,(-0.173294+\alpha)\,(0.098311+0.41033\alpha+\alpha^2) \end{pmatrix}$$

$d\phi/d\alpha = \nabla f(xk1).d=0 \to 214.251\,(-0.188279+\alpha)\,(0.0881986+0.342583\alpha+\alpha^2) == 0$

$\alpha \to 0.188279$

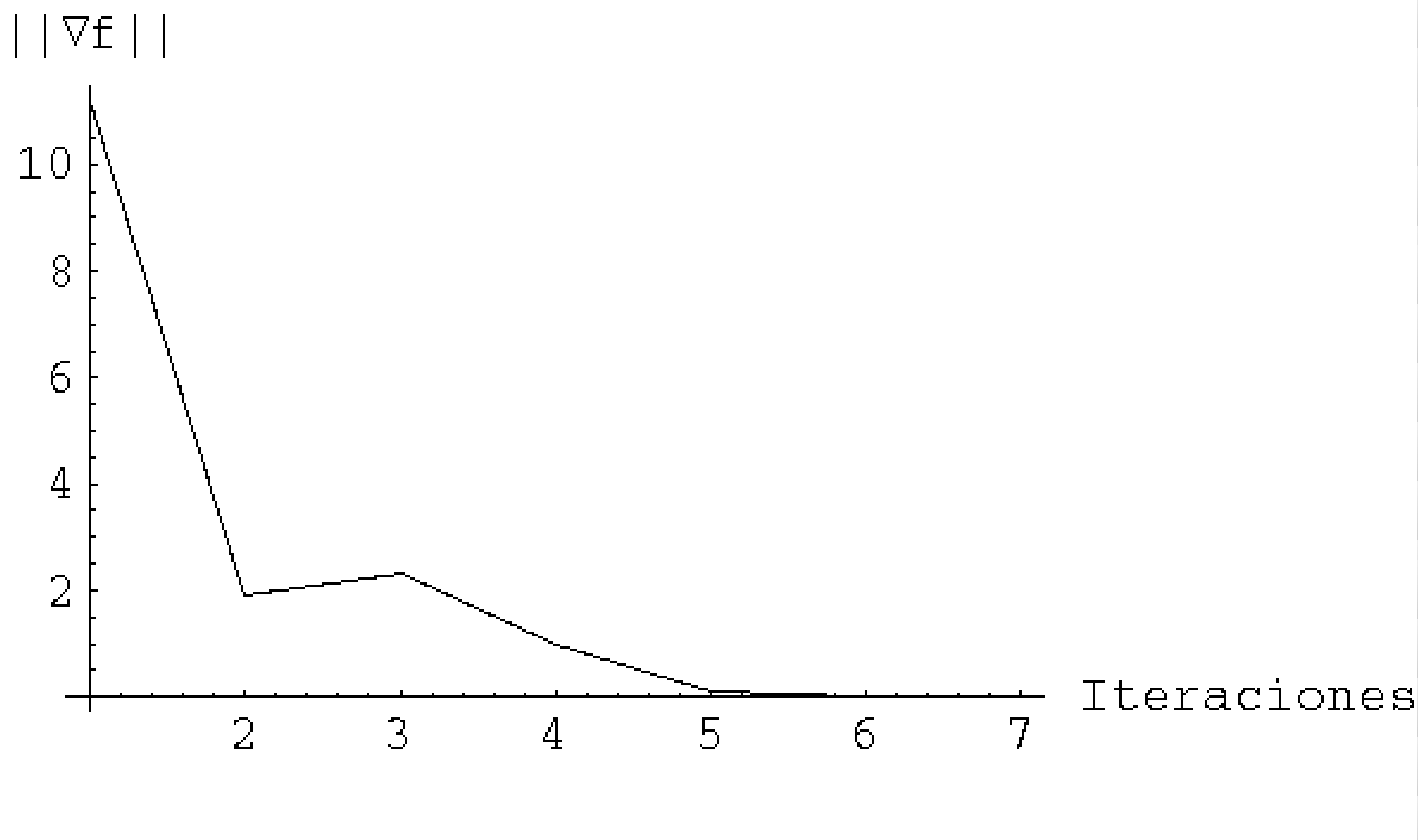
New Point (NonOptimum): [-0.948898, 0.60967] after 2 iterations

The second step is already different:
β=0.03 (Polak-Ribiere formula)



Note how the zigzag
Approach to the
Minimum is no
Longer true.

Note how the gradient norm, as a convergence measure, decreases
More steeply than the steepest descent: ~5 iterations vs ~20



- Multidimensional Newton method.
  - Start by using a second order Taylor series:

$$f(\boldsymbol{x}^{k+1}) \sim f(\boldsymbol{x}^{k}) + \nabla f(\boldsymbol{x}^{k})^{T} \boldsymbol{d}^{k} + \frac{1}{2} (\boldsymbol{d}^{k})^{T} \boldsymbol{H}(\boldsymbol{x}^{k}) \boldsymbol{d}^{k}$$

To find the minimum with respect to displacements ***d***:

$$\frac{d\, f(\boldsymbol{d}^{k})}{d\, \boldsymbol{d}^{k}} = \nabla f(\boldsymbol{x}^{k})^{T} + \boldsymbol{H}(\boldsymbol{x}^{k}) \boldsymbol{d}^{k} = \boldsymbol{0}$$

This equation can be solved for ***d***, its direct use leads to an unstable method. It is modified by an α value instead:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} + \alpha_{k} \boldsymbol{d}^{k} \quad k = 0, 1, \cdots$$

Where α is obtained by minimizing $f(\boldsymbol{x}^{k} + \alpha_{k} \boldsymbol{d}^{k})$

  - This method, like the one dimensional case, is much faster. However, it requires to do much more calculations, since, to find out ***d*** the Hessian has to be calculated and the system of equations solved.

- Example: Multidimensional Newton, same function.

$$f \to (x+y)^2 + \left(-\frac{1}{3} + 2\left(-1 + x^2 + y^2\right)\right)^2$$

$$\nabla f \to \begin{pmatrix} -\frac{50x}{3} + 16x^3 + 2y + 16xy^2 \\ 2x - \frac{50y}{3} + 16x^2y + 16y^3 \end{pmatrix}$$

$$\nabla^2 f \to \begin{pmatrix} -\frac{50}{3} + 48x^2 + 16y^2 & 2 + 32xy \\ 2 + 32xy & -\frac{50}{3} + 16x^2 + 48y^2 \end{pmatrix}$$

***** Iteration 1 ***** Current point $\to$ {-1.25, 0.25}

Direction-finding phase:

1st iteration:

$$\nabla^2 f \to \begin{pmatrix} 59.3333 & -8. \\ -8. & 11.3333 \end{pmatrix} \quad \nabla f(x) \to \begin{pmatrix} -11.1667 \\ -0.166667 \end{pmatrix}$$

$$\|\nabla f(x)\| \to 11.1679 \quad f(x) \to 1.84028$$

$$d \to \begin{pmatrix} 0.21019 & 0.163075 \end{pmatrix}$$

Step-length calculation phase:

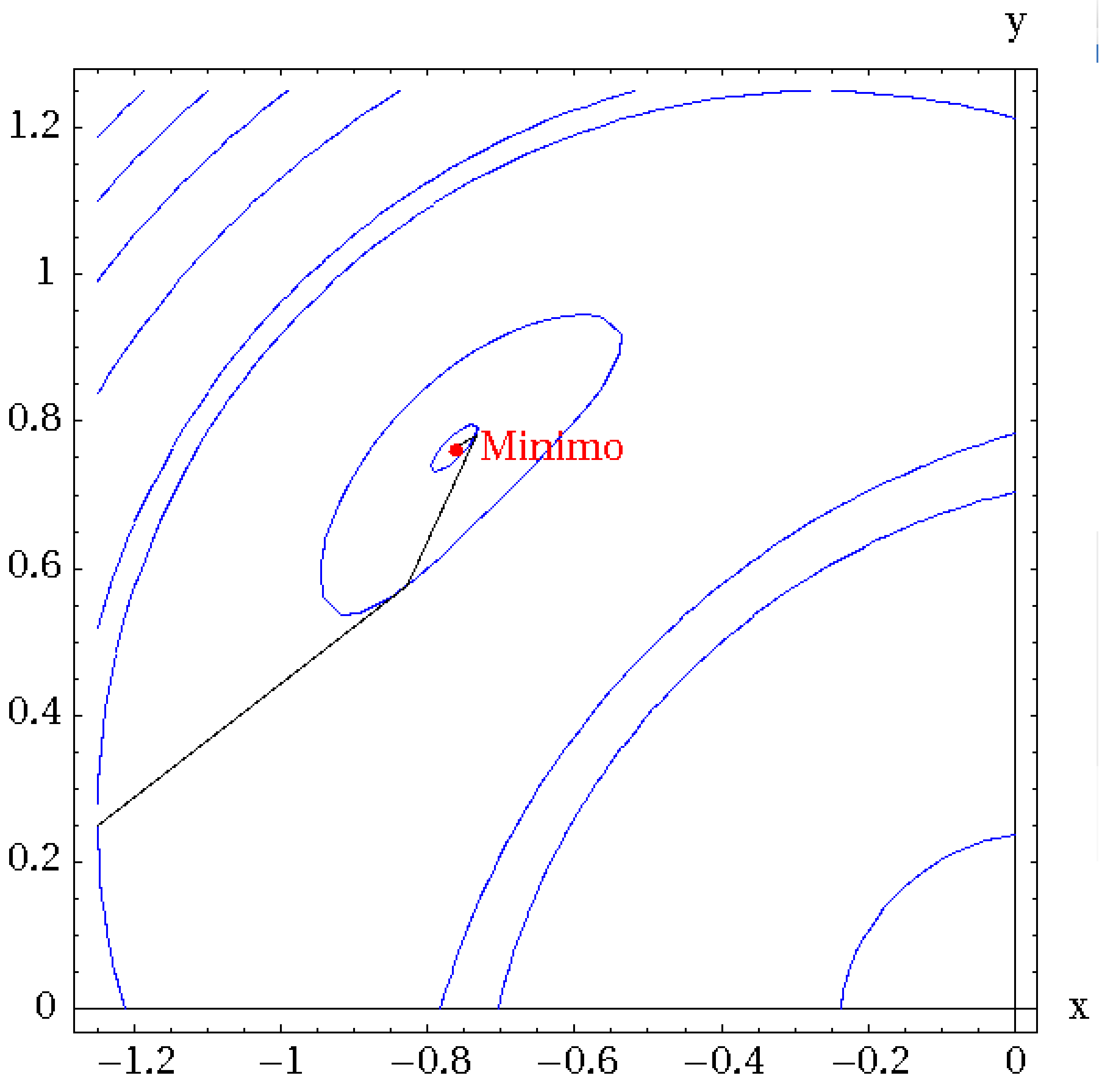
$$xk1 \to \begin{pmatrix} -1.25 + 0.21019\alpha \\ 0.25 + 0.163075\alpha \end{pmatrix}$$

$$\nabla f(xk1) \to \begin{pmatrix} 0.238013(-1.53958 + \alpha)(30.4732 - 10.6801\alpha + \alpha^2) \\ 0.184662(-4.58573 + \alpha)(0.196817 - 0.153898\alpha + \alpha^2) \end{pmatrix}$$

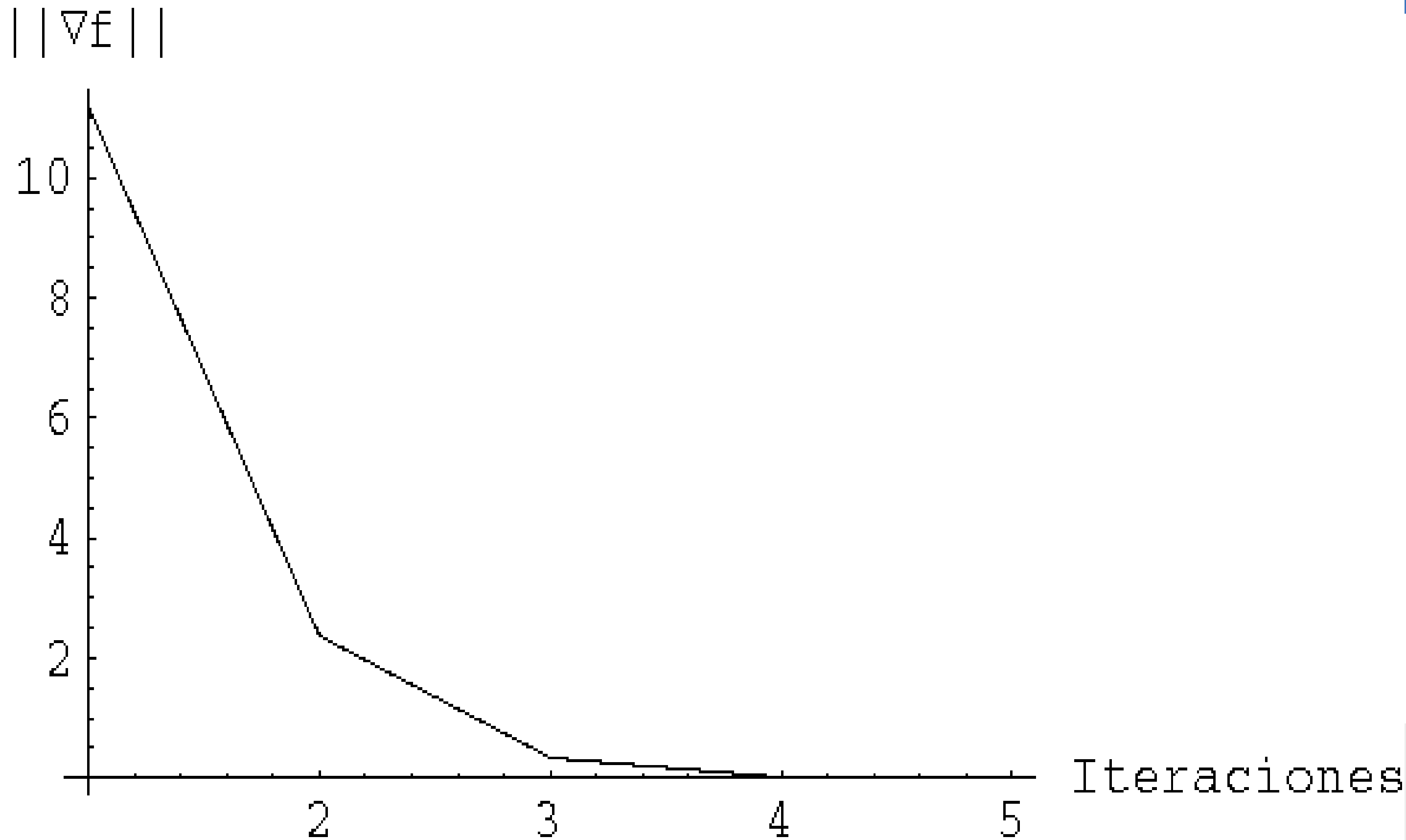
$$d\phi/d\alpha = \nabla f(xk1).d = 0 \to 0.0801418(-2.00245 + \alpha)(14.795 - 7.40654\alpha + \alpha^2) == 0$$

$$\alpha \to 2.00245$$

Multidimensional Newton Method.



The Newton Method is quite fast, but each iteration is very costly.



- Quasi-Newton Methods.
  - Keep the convergence speed of Newton methods without the penalty of heavy calculations at each step.
    - Formally, the step ***d*** in Newton's method is:

$$\nabla f(\boldsymbol{x}^k)^T + \boldsymbol{H}(\boldsymbol{x}^k)\,\boldsymbol{d}^k = \boldsymbol{0} \rightarrow \boldsymbol{d}^k = -\left(\boldsymbol{H}(\boldsymbol{x}^k)\right)^{-1} \nabla f(\boldsymbol{x}^k)$$

    - Instead of using it, an easier to calculate ***d*** is used:
      (1) Determine $\boldsymbol{d}^k = -\boldsymbol{Q}^k \nabla f(\boldsymbol{x}^k)$ *usually* $\boldsymbol{Q}_{inicial} = \boldsymbol{1}$
      (2) Calculate α and new point $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$
      (3) Calculate ***Q*** of the next iteration.
    - The ***Q*** matrix is calculated such that it approximates the inverse of the Hessian ***H***. There are two commonly used formulas: DFP y BFGS.

– DFP: Davidon, Fletcher y Powell.

$$\boldsymbol{Q}^{k+1} = \boldsymbol{Q}^k + \frac{\boldsymbol{s}^k (\boldsymbol{s}^k)^T}{(\boldsymbol{q}^k)^T \boldsymbol{s}^k} - \frac{(\boldsymbol{Q}^k \boldsymbol{q}^k)(\boldsymbol{Q}^k \boldsymbol{q}^k)^T}{(\boldsymbol{q}^k)^T \boldsymbol{Q}^k \boldsymbol{q}^k}$$

con:

$$\boldsymbol{q}^k = \nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^k)$$

$$\boldsymbol{s}^k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k \equiv \alpha_k \boldsymbol{d}^k$$

– BFGS: Broyden, Fletcher, Goldfarb y Shanon.

$$\boldsymbol{Q}^{k+1} = \boldsymbol{Q}^k + \left(1 + \frac{(\boldsymbol{q}^k)^T \boldsymbol{Q}^k \boldsymbol{q}^k}{(\boldsymbol{q}^k)^T \boldsymbol{s}^k}\right) \frac{\boldsymbol{s}^k (\boldsymbol{s}^k)^T}{(\boldsymbol{q}^k)^T \boldsymbol{s}^k} - \frac{1}{(\boldsymbol{q}^k)^T \boldsymbol{s}^k} \left[\left(\boldsymbol{s}^k (\boldsymbol{q}^k)^T \boldsymbol{Q}^k \right)^T + \boldsymbol{s}^k (\boldsymbol{q}^k)^T \boldsymbol{Q}^k \right]$$

The new ***Q*** is calculated only if $(\boldsymbol{q}^k)^T \boldsymbol{s}^k > 0$ Moreover, ***Q*** is set equal to the identity every **D** iterations, with **D** the dimension of the minimization space.

- Example: Quasi-Newton, same function.

$$f \to (x+y)^2 + \left(-\frac{1}{3} + 2\,(-1 + x^2 + y^2)\right)^2$$

$$\nabla f \to \begin{pmatrix} -\frac{50x}{3} + 16x^3 + 2y + 16xy^2 \\ 2x - \frac{50y}{3} + 16x^2y + 16y^3 \end{pmatrix}$$

Using the DFP method with approximate inverse
hessian reset after 10 iterations and Analytical line search

***** Iteration 1 ***** Current point $\to$ [-1.25, 0.25]

Direction-finding phase:

1st iteration:

Inverse Hessian $\to \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $\quad \nabla f(x) \to \begin{pmatrix} -11.1667 \\ -0.166667 \end{pmatrix}$

$\|\nabla f(x)\| \to 11.1679 \quad f(x) \to 1.84028$

$d \to (11.1667 \quad 0.166667)$

Step-length calculation phase:

FindRoot::bdmtd : Value of option Method -> DFP is not Automatic, Brent, Secant, or Newton. More...

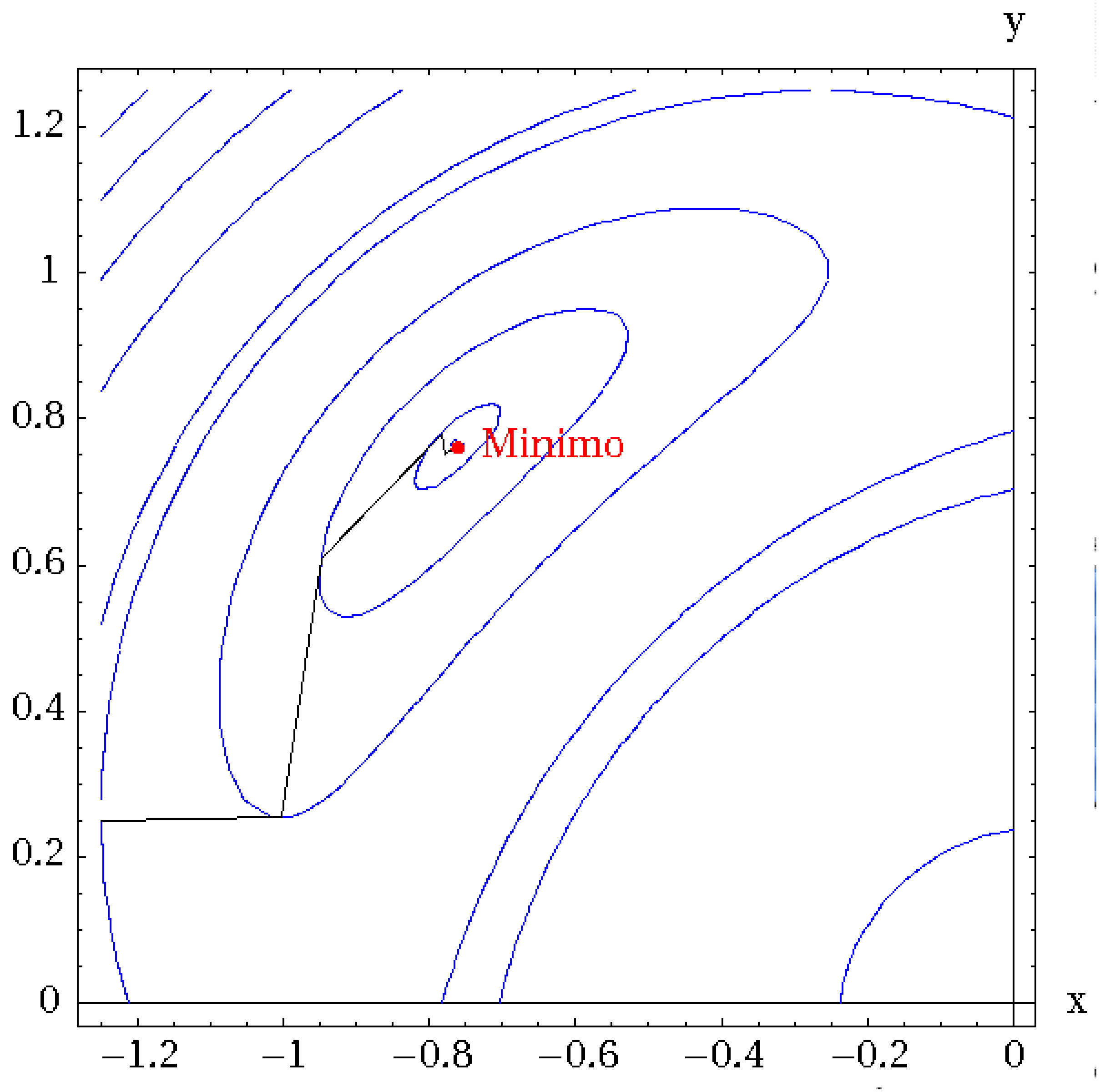
$$xk1 \to \begin{pmatrix} -1.25 + 11.1667\alpha \\ 0.25 + 0.166667\alpha \end{pmatrix}$$

$$\nabla f(xk1) \to \begin{pmatrix} 22283.7\,(-0.198059+\alpha)\,(-0.115053+\alpha)\,(-0.0219909+\alpha) \\ 332.593\,(-0.182098+\alpha)\,(0.00188867+\alpha)\,(1.45705+\alpha) \end{pmatrix}$$

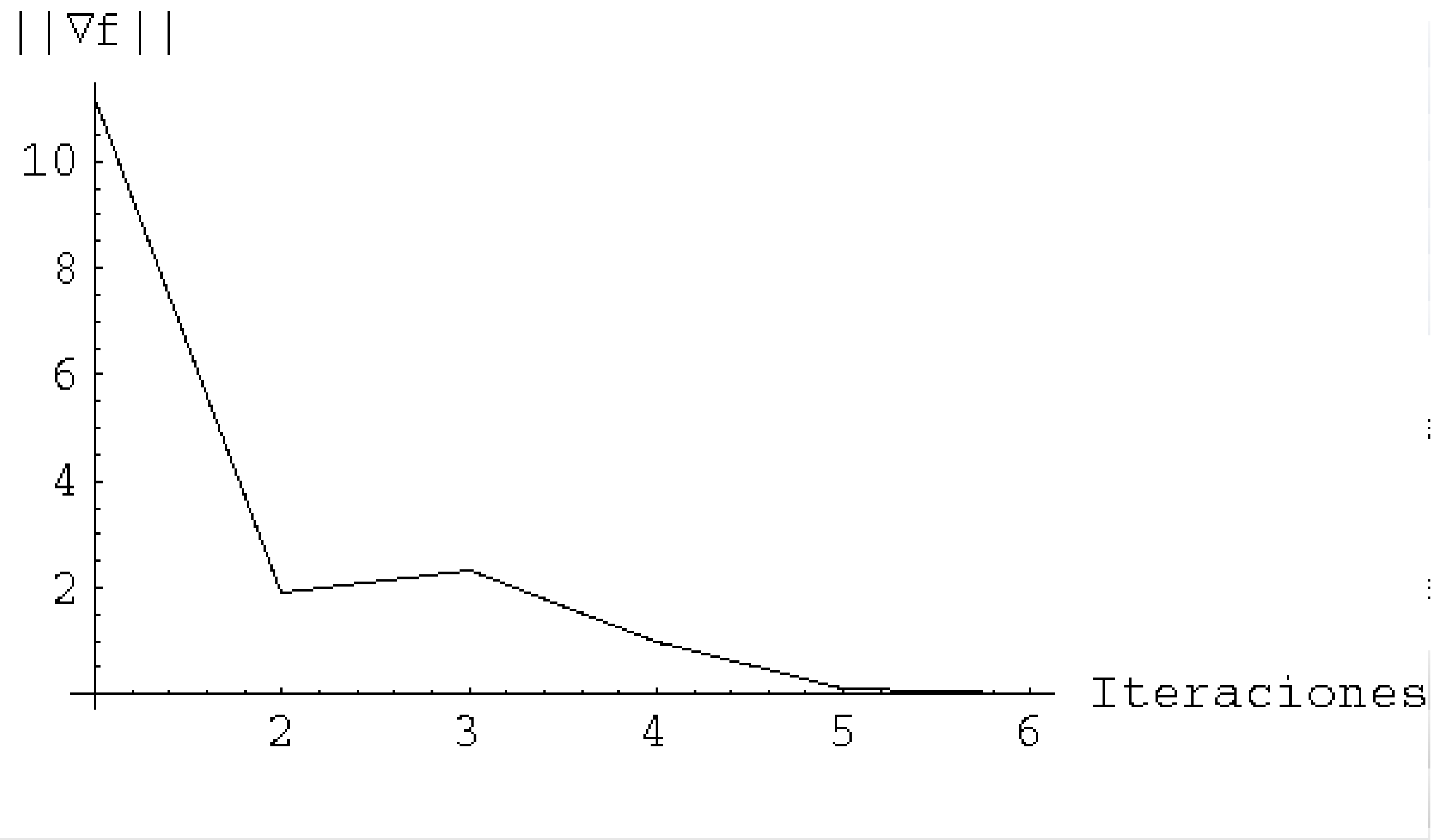
$d\phi/d\alpha = \nabla f(xk1).d=0 \to 248890.\,(-0.197979+\alpha)\,(-0.114697+\alpha)\,(-0.0220681+\alpha) == 0$

$\alpha \to 0.$

Quasi-Newton Method



The Quasi-Newton method is quite fast, although not as much as the modified Newton method. However, each iteration is much less costly.



Direct Search Methods

- Using a derivative is usually troublesome.
  - Numerically unstable.
  - Useless if the system is noisy.
  - Costly, specially if there is no direct access to the function.
- There are robust methods with known convergence properties that do not need a derivative (zeroth-order method).

Compass Search: A Generating Set Search Example

Initialization.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be given.

Let $x_0 \in \mathbb{R}^n$ be the initial guess.

Let $\Delta_{\mathrm{tol}} > 0$ be the tolerance used to test for convergence.

Let $\Delta_0 > \Delta_{\mathrm{tol}}$ be the initial value of the step-length control parameter.

Algorithm. For each iteration $k = 1, 2, \ldots$

Step 1. Let $\mathcal{D}_\oplus$ be the set of coordinate directions $\{\pm e_i \,|\, i = 1, \ldots, n\}$, where e_i is the ith unit coordinate vector in $\mathbb{R}^n$.

Step 2. If there exists $d_k \in \mathcal{D}_\oplus$ such that $f(x_k + \Delta_k d_k) < f(x_k)$, then do the following:

- Set $x_{k+1} = x_k + \Delta_k d_k$ (change the iterate).
- Set $\Delta_{k+1} = \Delta_k$ (no change to the step-length control parameter).

Step 3. Otherwise, $f(x_k + \Delta_k d) \geq f(x_k)$ for all $d \in \mathcal{D}_\oplus$, so do the following:

- Set $x_{k+1} = x_k$ (no change to the iterate).
- Set $\Delta_{k+1} = \frac{1}{2}\Delta_k$ (contract the step-length control parameter).
- If $\Delta_{k+1} < \Delta_{\mathrm{tol}}$, then **terminate**.

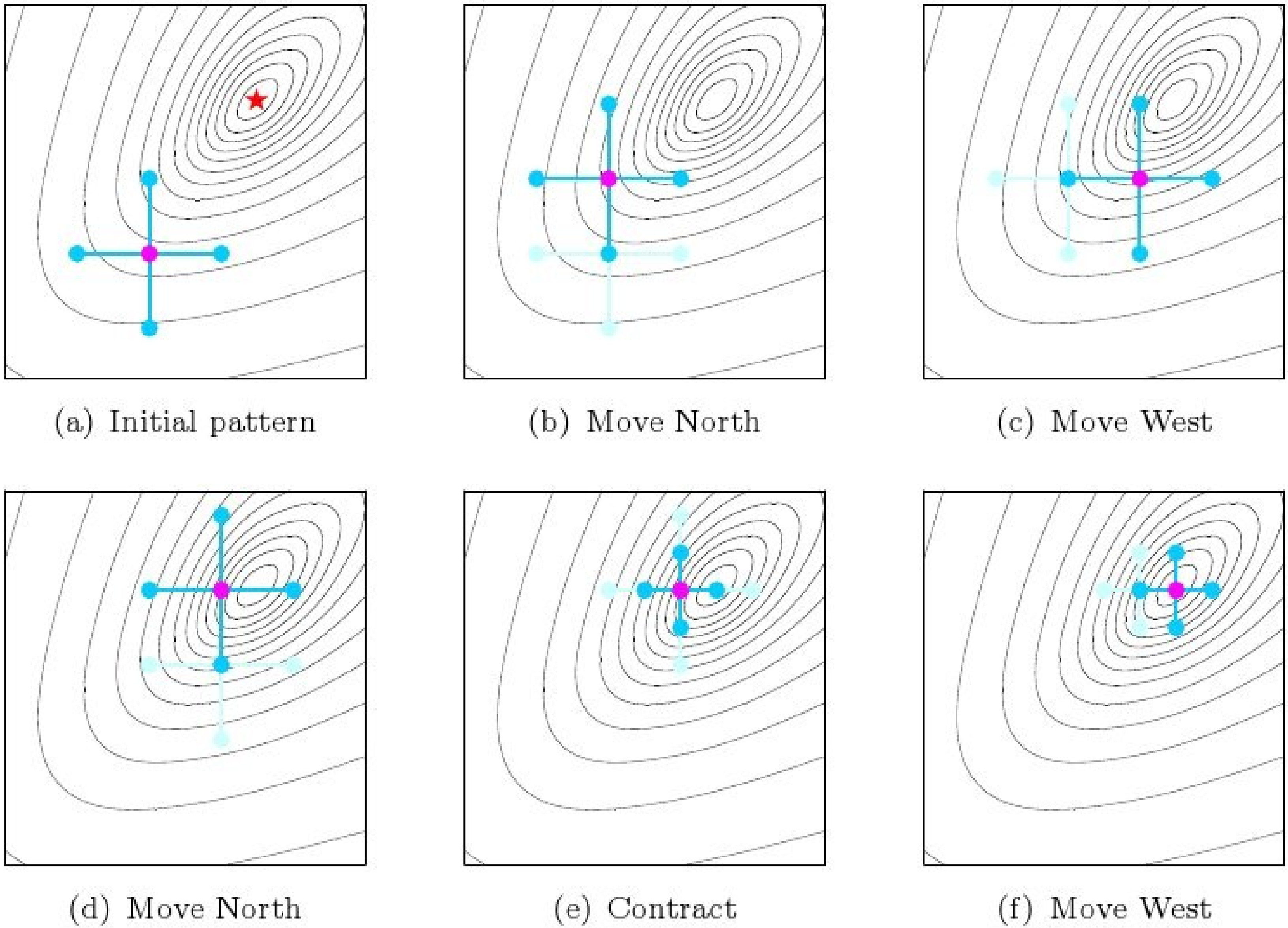


Fig. 1.1 *Compass search applied to the modified Broyden tridiagonal function.*

KLT03

Generating Set Search Algorithm: Initialization.

Initialization.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be given.

Let $x_0 \in \mathbb{R}^n$ be the initial guess.

Let $\Delta_{\text{tol}} > 0$ be the step-length convergence tolerance.

Let $\Delta_0 > \Delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\theta_{\max} < 1$ be an upper bound on the contraction parameter.

Let $\rho : [0, +\infty) \to \mathbb{R}$ be a continuous function such that $\rho(t)$ is decreasing as $t \to 0$ and $\rho(t)/t \to 0$ as $t \downarrow 0$. The choice $\rho \equiv 0$ is acceptable.

Let $\beta_{\max} \geq \beta_{\min} > 0$ be upper and lower bounds, respectively, on the lengths of the vectors in any generating set.

Let $\kappa_{\min} > 0$ be a lower bound on the cosine measure of any generating set.

KLT03

Algorithm. For each iteration $k = 1, 2, \ldots$

Step 1. Let $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$. Here $\mathcal{G}_k$ is a generating set for $\mathbb{R}^n$ satisfying $\beta_{\min} \leq \|d\| \leq \beta_{\max}$ for all $d \in \mathcal{G}_k$ and $\kappa(\mathcal{D}_k) \geq \kappa_{\min}$, and $\mathcal{H}_k$ is a finite (possibly empty) set of additional search directions such that $\beta_{\min} \leq \|d\|$ for all $d \in \mathcal{H}_k$.

Step 2. If there exists $d_k \in \mathcal{D}_k$ such that $f(x_k + \Delta_k d_k) < f(x_k) - \rho(\Delta_k)$, then do the following:

- Set $x_{k+1} = x_k + \Delta_k d_k$ (change the iterate).
- Set $\Delta_{k+1} = \phi_k \Delta_k$, where $\phi_k \geq 1$ (optionally expand the step-length control parameter).

Step 3. Otherwise, $f(x_k + \Delta_k d) \geq f(x_k) - \rho(\Delta_k)$ for all $d \in \mathcal{D}_k$, so do the following:

- Set $x_{k+1} = x_k$ (no change to the iterate).
- Set $\Delta_{k+1} = \theta_k \Delta_k$ where $0 < \theta_k < \theta_{\max} < 1$ (contract the step-length control parameter).
- If $\Delta_{k+1} < \Delta_{\text{tol}}$, then **terminate**.

KLT03

Generating Set Search Algorithm.

Generating Set Search Algorithm.

- Slow... but sure.
- Performance decreases with increased space dimension.

Bibliography.

- M.A. Bhatti. Practical Optimization Methods. Springer-Verlag, 2000. and programs therein.
- A.L. Peressini et al. The Mathematics of Nonlinear Programming. Springer-Verlag, 1988
- Nocedal
- Kolda, Lewis, Torczon. Optimization by Direct Search. SIAM Review 45, 385-482, 2003.