MCCSN
TEMA 4

SERGIO PERNICE

UCEMA

OPTIMIZACIÓN SIN RESTRICCIONES: SERIE DE TAYLOR

- 1-D Serie de Taylor, o como aproximar una función f(x) real en la vecindad de un punto “a” (se asume conocido en 1-D):

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{2!}(x - x_0)^3 + \cdots$$

- Ejemplo: f(x) = sen(x), x_0 = 0: $f^{(1)}$(x) = cos(x), $f^{(2)}$(x) = -sen(x), $f^{(3)}$(x) = -cos(x), $f^{(4)}$(x) = sen(x),…
  - f(0) = 0, $f^{(1)}$(0) = 1, $f^{(2)}$(0) = 0 , $f^{(3)}$(0) = -1 , $f^{(4)}$(0) = 0

$$f(x) = 0 + x + 0 - \frac{x^3}{3!} + \cdots = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$

- https://www.geogebra.org/m/C4S6CEdm

$$
\begin{aligned}
f(x_1,\ldots,x_d) \;&=\; f(a_1,\ldots,a_d) + \sum_{j=1}^{d} \frac{\partial f(a_1,\ldots,a_d)}{\partial x_j}\left(x_j - a_j\right) \\
&+ \frac{1}{2!}\sum_{j=1}^{d}\sum_{k=1}^{d} \frac{\partial^2 f(a_1,\ldots,a_d)}{\partial x_j \partial x_k}\left(x_j - a_j\right)(x_k - a_k) + \cdots
\end{aligned}
$$

OPTIMIZACIÓN SIN RESTRICCIONES: SERIE DE TAYLOR

- d-D Serie de Taylor, o como aproximar una función f(**x**): $R^d \rightarrow R$ en la vecindad de un punto "**a**"
- Para visualizar el caso d = 2 ver: https://www.geogebra.org/m/chMpCQz2

OPTIMIZACIÓN SIN RESTRICCIONES: SERIE DE TAYLOR

$$f(x_1,\ldots,x_d) = f(a_1,\ldots,a_d) + \sum_{j=1}^{d} \frac{\partial f(a_1,\ldots,a_d)}{\partial x_j}\left(x_j - a_j\right)$$
$$+\frac{1}{2!}\sum_{j=1}^{d}\sum_{k=1}^{d} \frac{\partial^2 f(a_1,\ldots,a_d)}{\partial x_j \partial x_k}\left(x_j - a_j\right)(x_k - a_k) + \cdots$$

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{a}) + \nabla_{\mathbf{x}} f(\mathbf{a}) \cdot \mathbf{y} + \frac{1}{2}\mathbf{y}^{\mathrm{T}} \cdot \nabla_{\mathbf{x}}^2 f(\mathbf{a}) \cdot \mathbf{y} + \cdots \\ \mathbf{y} &= \mathbf{x} - \mathbf{a} \end{aligned}$$

$$\nabla_{\mathbf{x}} f(\mathbf{a}) = \begin{pmatrix} \partial_{x_1} f \\ \partial_{x_2} f \\ \vdots \\ \partial_{x_n} f \end{pmatrix}(\mathbf{a}) = \text{gradient}$$

$$\nabla_{\mathbf{x}}^2 f(\mathbf{a}) = \begin{pmatrix} \partial^2_{x_1,x_1} f & \partial^2_{x_1,x_2} f & \cdots & \partial^2_{x_1,x_n} f \\ \partial^2_{x_2,x_1} f & \partial^2_{x_2,x_2} f & \cdots & \partial^2_{x_2,x_n} f \\ \vdots & \vdots & \ddots & \vdots \\ \partial^2_{x_n,x_1} f & \partial^2_{x_n,x_2} f & \cdots & \partial^2_{x_n,x_n} f \end{pmatrix}(\mathbf{a}) = \text{Hessian}$$

APLICACIONES 1: PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (ALPHA-CHIANG, CAP. 11)

- Empresa con dos productos, competencia perfecta (precios son variable exógena)
- Ventas: $R = p_1 q_1 + p_2 q_2$
- Costos: $C = 2 q_1^2 + q_1 q_2 + 2 q_2^2$
- Ganancia: $\pi = R - C$
- Expresar π en términos de vectoriales y matriciales (gradiente y Hessiano)

APLICACIONES 1: PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (ALPHA-CHIANG, CAP. 11)

- Resolvamos primero el problema de una empresa en competencia perfecta con 1 solo producto, y costo cuadrático de producción.
- Pensar en el significado geométrico del cambio de variables de q a x

$$
\begin{aligned}
\Pi &= pq - \frac{1}{2}cq^2 = pq - \frac{1}{2}qcq \\
\frac{d\Pi}{dq} &= p - cq_{\min} = 0 \\
cq_{\min} &= p \\
q_{\min} &= c^{-1}p
\end{aligned}
$$

$$
q = q_{\min} + x = c^{-1}p + x
$$

$$
\begin{aligned}
\Pi &= p\left(c^{-1}p + x\right) - \frac{1}{2}\left(pc^{-1} + x\right)c\left(c^{-1}p + x\right) \\
&= pc^{-1}p + px - \frac{1}{2}\left(pc^{-1}cc^{-1}p + pc^{-1}cx + xcc^{-1}p + xcx\right) \\
&= \frac{1}{2}pc^{-1}p - \frac{1}{2}xcx
\end{aligned}
$$

Notar que el resultado final es una constante (el valor de π en el extremo) más una forma cuadrática

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO
BREVE INTERLUDIO SOBRE ALGEBRA DE MATRICES

- $(AB)^T = B^T A^T$
- Si C simétrica y con inversa: $I = I^T = (C\,C^{-1})^T = (C^{-1})^T\,C^T = (C^{-1})^T\,C = I$ ⍰ $(C^{-1})^T = C^{-1}$, es decir, C^{-1} también es simétrica.
- Si U es una matriz ortogonal, sus columnas forman una base ortonormal, lo mismo ocurre con sus filas, y su inversa es la traspuesta: $U^{-1} = U^T$.
- Si **p** y **q** son vectores (o matrices n x 1):

$$\boldsymbol{p} \cdot \boldsymbol{q} = \boldsymbol{q} \cdot \boldsymbol{p} = \boldsymbol{p}^T \boldsymbol{q} = \boldsymbol{q}^T \boldsymbol{p}$$

- Para matrices A y B, en general $AB \neq BA$ (si ambas multiplicaciones se puedan realizar)

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO
BREVE INTERLUDIO SOBRE DERIVADAS EN FORMATO VECTORIZADO (C ES SIMÉTRICA)

$$
\begin{aligned}
f(\mathbf{x}) &= \mathbf{x}^{\mathrm{T}}\mathbf{a} = \sum_{i=1}^{n} x_i a_i \\
\frac{\partial f}{\partial x_k} &= a_k \\
\nabla f(\mathbf{x}) &= \mathbf{a}
\end{aligned}
$$

$$
\begin{aligned}
g(\mathbf{x}) &= \frac{1}{2}\mathbf{x}^{\mathrm{T}} C\,\mathbf{x} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_i x_j \\
\frac{\partial g}{\partial x_k} &= \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_i \delta_{jk} + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\delta_{ik} x_j\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_j \\
\nabla g(\mathbf{x}) &= C\,\mathbf{x}
\end{aligned}
$$

Notar la similitud formal con el caso de 1 variable: f = x a, f’ = a, g = c $x^2/2$ = x c x/2 , g’ = c x

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (U OPTIMIZACIÓN DE UNA FUNCIÓN GENÉRICA DE ORDEN 2)

$$
\begin{aligned}
\Pi &= \mathbf{p}^{\mathrm{T}}\mathbf{q} - \frac{1}{2}\mathbf{q}^{\mathrm{T}}C\mathbf{q} \\
\nabla\Pi &= \mathbf{p} - C\mathbf{q}_{\min} = \mathbf{0} \\
C\mathbf{q}_{\min} &= \mathbf{p} \\
\mathbf{q}_{\min} &= C^{-1}\mathbf{p}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{q} &= \mathbf{q}_{\min} + \mathbf{x} = C^{-1}\mathbf{p} + \mathbf{x} \\
\Pi &= \mathbf{p}^{\mathrm{T}}\left(C^{-1}\mathbf{p} + \mathbf{x}\right) - \frac{1}{2}\left(\mathbf{p}^{\mathrm{T}}C^{-1} + \mathbf{x}^{\mathrm{T}}\right)C\left(C^{-1}\mathbf{p} + \mathbf{x}\right) \\
&= \mathbf{p}^{\mathrm{T}}C^{-1}\mathbf{p} + \mathbf{p}^{\mathrm{T}}\mathbf{x} - \frac{1}{2}\left(\mathbf{p}^{\mathrm{T}}C^{-1}CC^{-1}\mathbf{p} + \mathbf{p}^{\mathrm{T}}C^{-1}C\mathbf{x} + \mathbf{x}^{\mathrm{T}}CC^{-1}\mathbf{p} + \mathbf{x}C\mathbf{x}\right) \\
&= \frac{1}{2}\mathbf{p}^{\mathrm{T}}C^{-1}\mathbf{p} - \frac{1}{2}\mathbf{x}^{\mathrm{T}}C\mathbf{x}
\end{aligned}
$$

- Pensar en el significado geométrico del cambio de variables de **q** a **x.**
- Notar que en notación vectorial, la optimización es formalmente idéntica a la de una variable, pero acá estamos en R^n para n cualquiera!

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (U OPTIMIZACIÓN DE UNA FUNCIÓN GENÉRICA DE ORDEN 2)

- La función π: $R^n \rightarrow R$, de orden 2 tiene un punto crítico (donde el gradiente es **0**) en $\mathbf{q} = C^{-1}\mathbf{p}$, y en ese punto es igual a $\mathbf{p}^T C^{-1}\mathbf{p}/2$.
- Conviene entonces trasladar el origen de coordenadas a dicho punto. Llamamos **x** a los vectores desde ese origen. El punto crítico corresponde entonces a **x** = **0.**
- Para analizar dicho punto conviene "rotar" el sistema de coordenadas al generado por la base de autovectores ortonormales de C, donde la forma cuadrática $\mathbf{x}^T C\mathbf{x}$ es diagonal.
- En ese sistema de coordenadas es obvio que ese punto va a ser un máximo, un mínimo o un punto "silla de montar" dependiendo de si los autovalores de C son < 0, > 0, o de ambos signos.

$$\Pi = \mathbf{p}^T\mathbf{q} - \frac{1}{2}\mathbf{q}^T C\mathbf{q}, \quad \mathbf{q} = \mathbf{q}_{\min} + \mathbf{x} = C^{-1}\mathbf{p} + \mathbf{x}, \quad \Pi = \frac{1}{2}\mathbf{p}^T C^{-1}\mathbf{p} - \frac{1}{2}\mathbf{x}^T C\mathbf{x}$$

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (U OPTIMIZACIÓN DE UNA FUNCIÓN GENÉRICA DE ORDEN 2)

- La única posibilidad no considerada aún es que haya autovalores nulos.
- Para analizar este caso conviene pausar y meditar sobre lo que hemos hecho.
- Quisimos analizar una función de la forma $\pi = \mathbf{p}^T \mathbf{q} - \mathbf{q}^T C \mathbf{q}/2$ y para eso primero encontramos el punto crítico, y luego hicimos dos operaciones geométricas:
  - Una traslación del origen al punto crítico para que este sea el nuevo origen. Desde ese origen la función es simplemente una constante más una forma cuadrática $\pi = \mathbf{p}^T C^{-1} \mathbf{p} - \mathbf{x}^T C \mathbf{x}/2$
  - Una transformación de la base canónica a la base de autovectores de C. En esta base la forma cuadrática es diagonal y su análisis es trivial si conocemos los autovalores. Dado que la base de autovectores también es ortonormal, esa transformación de una base a otra es una especie de "rotación" (más estrictamente, una "transformación ortogonal")
- Intuitivamente, hacer primero la traslación y después la "rotación" tiene que ser lo mismo que hacer primero la rotación y después la traslación. Pensar y tratar de intuir esto…
  - Dos rotaciones, en 3-D o más, no necesariamente conmutan, pero una rotación y una traslación siempre conmutan

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (U OPTIMIZACIÓN DE UNA FUNCIÓN GENÉRICA DE ORDEN 2)

- Realicemos entonces lo mismo que antes pero primero expresando los vectores en la base de autovectores ortonormales {$\mathbf{v}_j$} de C: C $\mathbf{v}_i = \lambda_i \mathbf{v}_i$ (recuerden que $\mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$)

$$
\begin{aligned}
\Pi &= \mathbf{p}^{\mathrm{T}}\mathbf{q} - \frac{1}{2}\mathbf{q}^{\mathrm{T}} C \mathbf{q} \\
\mathbf{q} &= \sum_{i=1}^{n} c_i \mathbf{v_i}, \qquad \mathbf{p} = \sum_{i=1}^{n} r_i \mathbf{v_i} \\
\mathbf{q}^{\mathrm{T}} &= \sum_{i=1}^{n} c_i \mathbf{v_i}^{\mathrm{T}}, \qquad \mathbf{p}^{\mathrm{T}} = \sum_{i=1}^{n} r_i \mathbf{v_i}^{\mathrm{T}} \\
\Pi &= \sum_{i=1}^{n} \left(r_i c_i - \frac{1}{2}\lambda_i c_i^2 \right) = -\sum_{i=1}^{n} \frac{1}{2}\lambda_i \left(c_i^2 - 2\frac{r_i}{\lambda_i} c_i \right) \\
&= \frac{1}{2}\sum_{i=1}^{n} \frac{r_i^2}{\lambda_i} - \frac{1}{2}\sum_{i=1}^{n} \lambda_i \left(c_i - \frac{r_i}{\lambda_i} \right)^2
\end{aligned}
$$

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (U OPTIMIZACIÓN DE UNA FUNCIÓN GENÉRICA DE ORDEN 2)

- Mirando fijo la ultima expresión, vemos que π en el punto extremo vale $\frac{1}{2}\sum_{i=1}^{n}\frac{r_i^2}{\lambda_i}$, donde r_i son las coordenadas de **p** en la base {$\mathbf{v}_i$} de autovectores de C. Pero en esa base C es diagonal con elementos diagonales λ_i, por lo que en la misma base C^{-1} también es diagonal elementos diagonales 1/ λ_i. Pero entonces en esa base $\frac{1}{2}\sum_{i=1}^{n}\frac{r_i^2}{\lambda_i} = \frac{1}{2}\boldsymbol{p}^T C^{-1}\boldsymbol{p}$ como habíamos descubierto antes!
- De la misma manera, si miramos fijo al segundo termino en π vemos que el mínimo corresponde a las coordenadas $c_i = \frac{r_i}{\lambda_i}$ en la base {$\mathbf{v}_i$} de autovectores de C, pero como dijimos, en esa base C^{-1} también es diagonal elementos diagonales 1/ λ_i, por lo que el mínimo en esa base se logra en el vector

$$\boldsymbol{q_{min}} = \sum_{i=1}^{n}\frac{r_i}{\lambda_i}\boldsymbol{v_i} = C^{-1}\boldsymbol{p}$$

pero esto también es el resultado que habíamos obtenido antes expresado en la base canónica!

PROBLEMA DE UNA EMPRESA MULTIPRODUCTO (U OPTIMIZACIÓN DE UNA FUNCIÓN GENÉRICA DE ORDEN 2)

- Entonces, del segundo término de π vemos que el punto $\mathbf{q}_{min} = C^{-1}\mathbf{p}$ será un mínimo, un máximo o un punto de silla de montar, dependiendo del signo de los autovalores de C, tal como habíamos entendido con formas cuadráticas.

- Y si algún autovalor es cero, entonces conviene mirar la primera expresión de π en la base de autovectores

$$\Pi = \sum_{i=1}^{n} \left(r_i c_i - \frac{1}{2} \lambda_i c_i^2 \right)$$

si algún λ_i es cero, entonces no tiene ni máximo ni mínimo. En el subespacio asociado a los autovectores con autovalor igual a cero π es una función lineal (notar que las operaciones subsiguientes a la expresión de más arriba se hacen dividiendo por λ_i, por lo que dichas operaciones son “ilegales” si λ_i es cero)

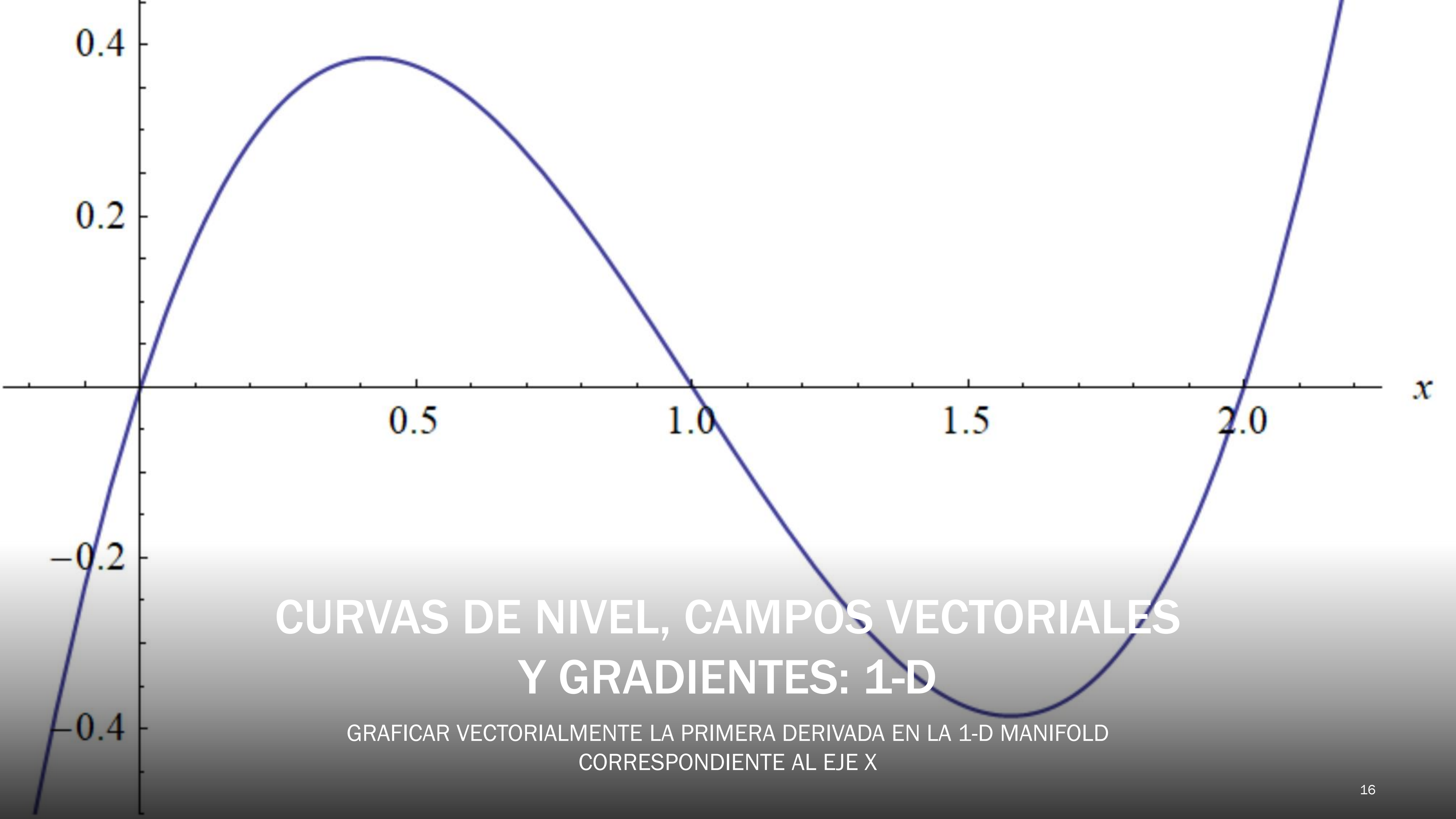
- Si se diera la casualidad de que **p** no tiene componentes en ese subespacio (los r_i correspondientes a la expansión de **p** en términos de los autovectores $\mathbf{v}_i$ de C son cero para los $\mathbf{v}_i$ correspondientes a $\lambda_i = 0$), entonces en esas direcciones π sería plana.

CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES

- Como vimos, la extensión a d-D de los criterios en 1-D para analizar puntos críticos son los siguientes:

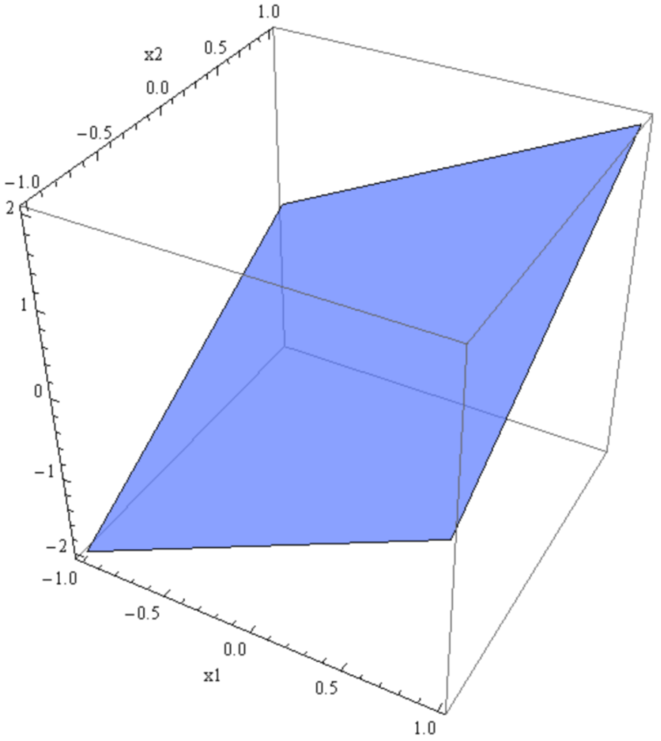
1-D	d-D
x tal que $df/dx = 0$	**x** tal que $\nabla f = 0$
$d^2f/dx^2 > 0 \Leftrightarrow$ mínimo	Autovalores de $\nabla^2 f > 0 \Leftrightarrow$ mínimo
$d^2f/dx^2 < 0 \Leftrightarrow$ máximo	Autovalores de $\nabla^2 f < 0 \Leftrightarrow$ máximo
	Autovalores de $\nabla^2 f$ > y < 0 $\Leftrightarrow$ punto silla de montar
$d^2f/dx^2 = 0 \Longrightarrow$ hay que analizar más	Algún autovalor de $\nabla^2 f = 0 \Longrightarrow$ hay que analizar más

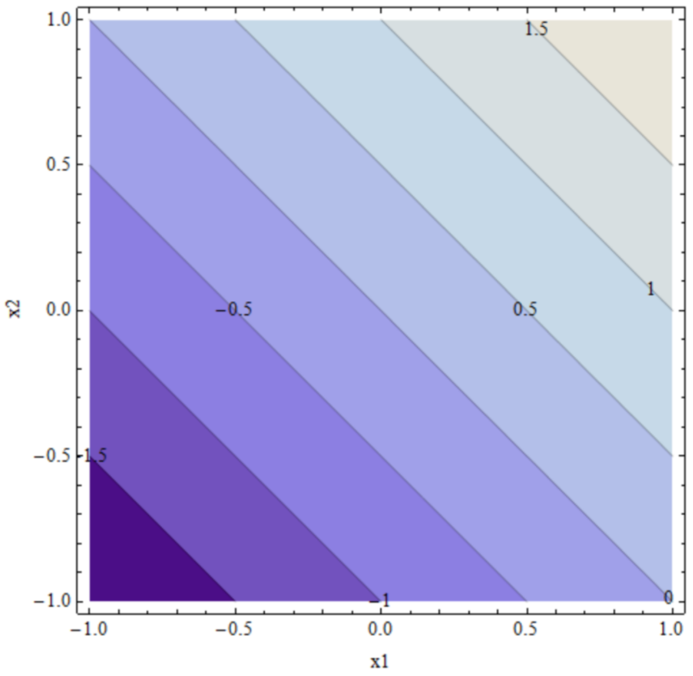
La derivada primera en 1-D, o el gradiente en d-D, en <u>todo</u> punto de la manifold define un campo vectorial en la misma, o una función $R^d \rightarrow R^d$ (en este caso el primer R^d se refiere a puntos de la manifold, y el segundo a vectores) que es de enorme utilidad.

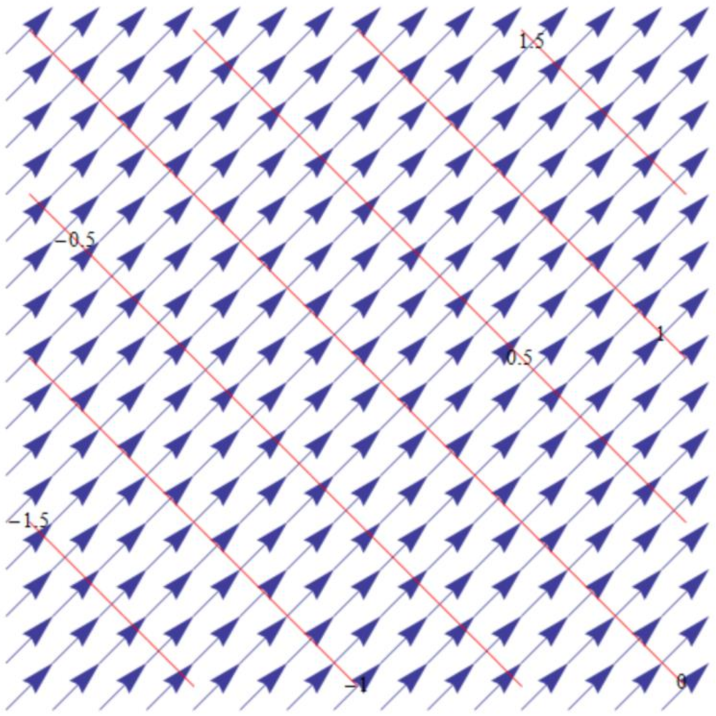


CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES: 1-D

GRAFICAR VECTORIALMENTE LA PRIMERA DERIVADA EN LA 1-D MANIFOLD CORRESPONDIENTE AL EJE X

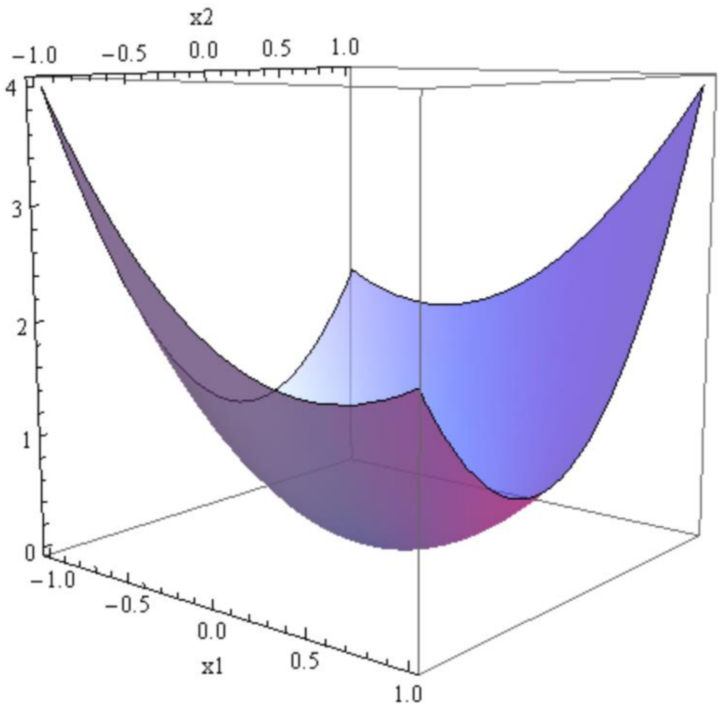


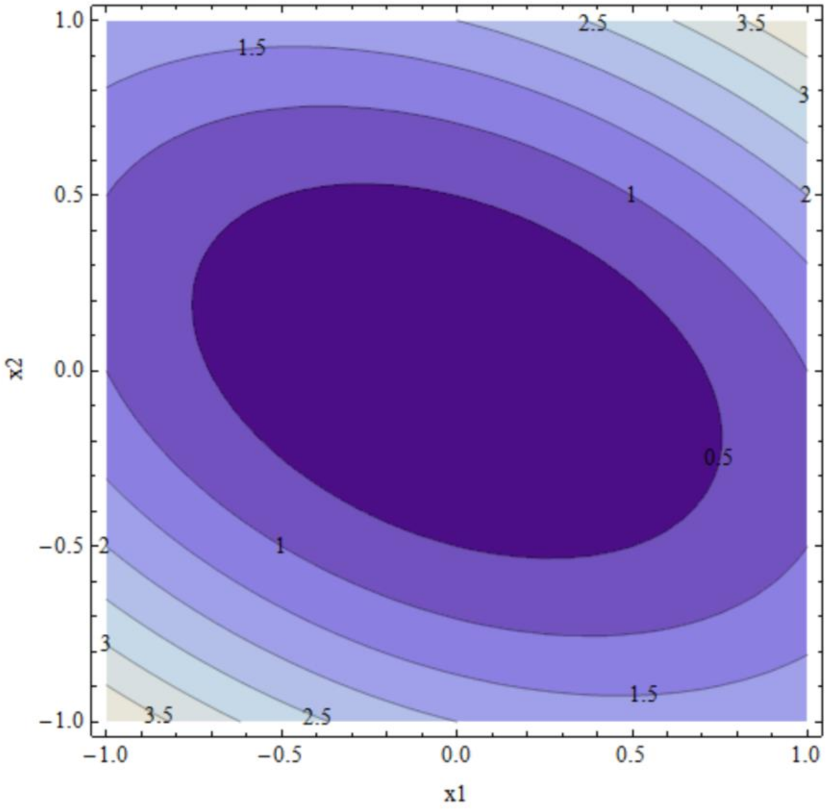


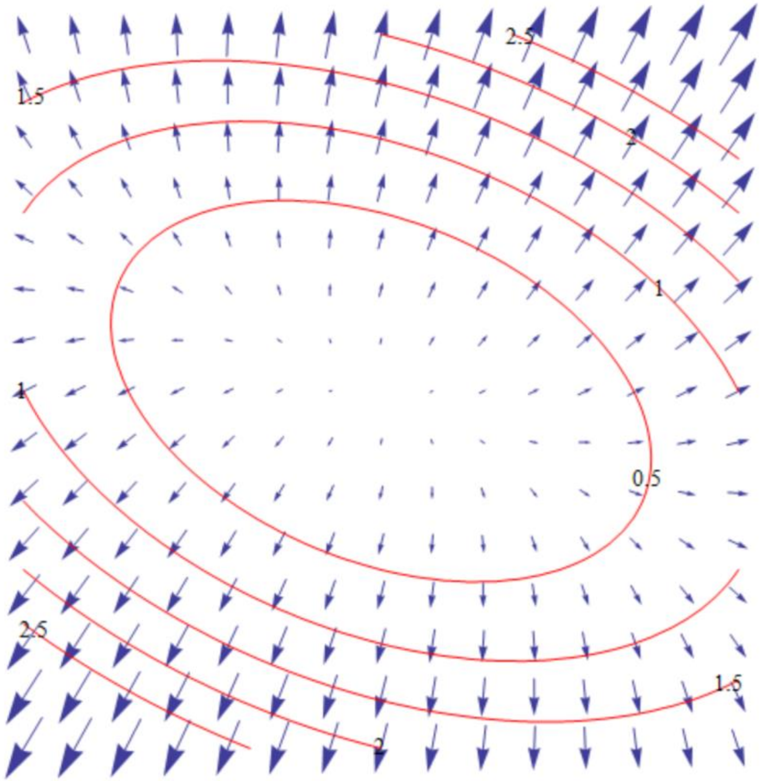


CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES: 2-D

- $f(x) = \boldsymbol{b}^T x, \ \boldsymbol{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \boldsymbol{\nabla} f = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

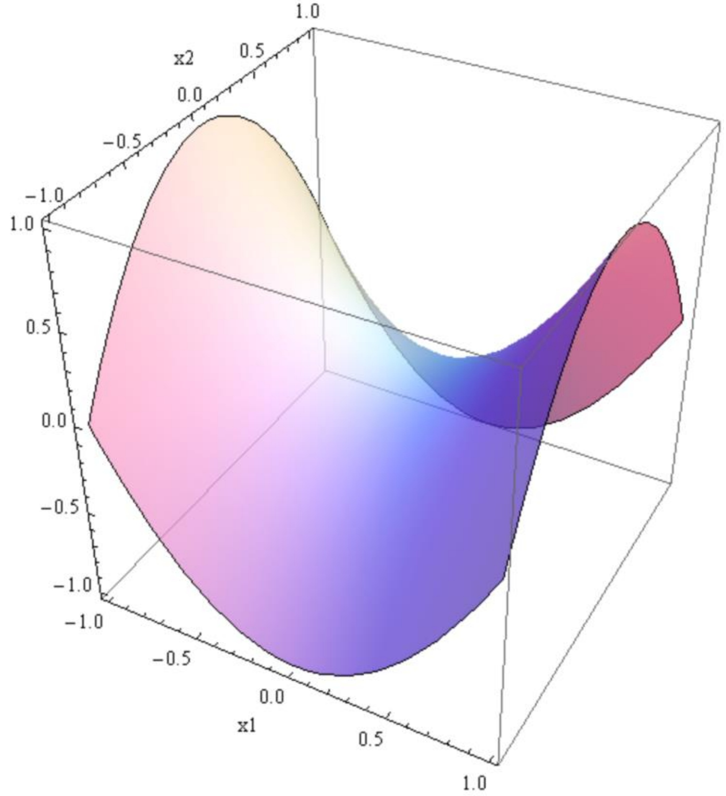


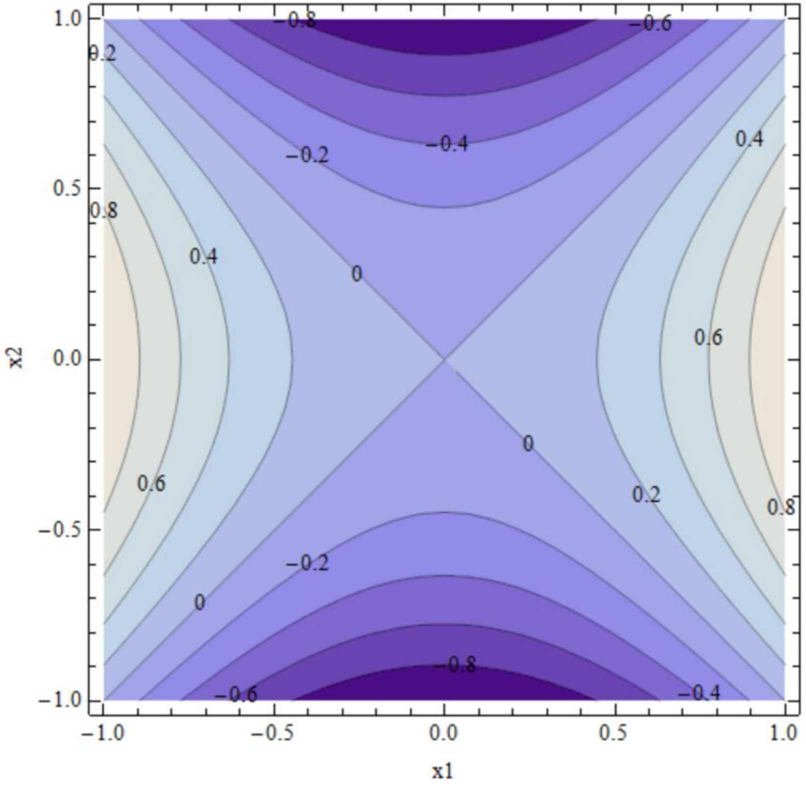


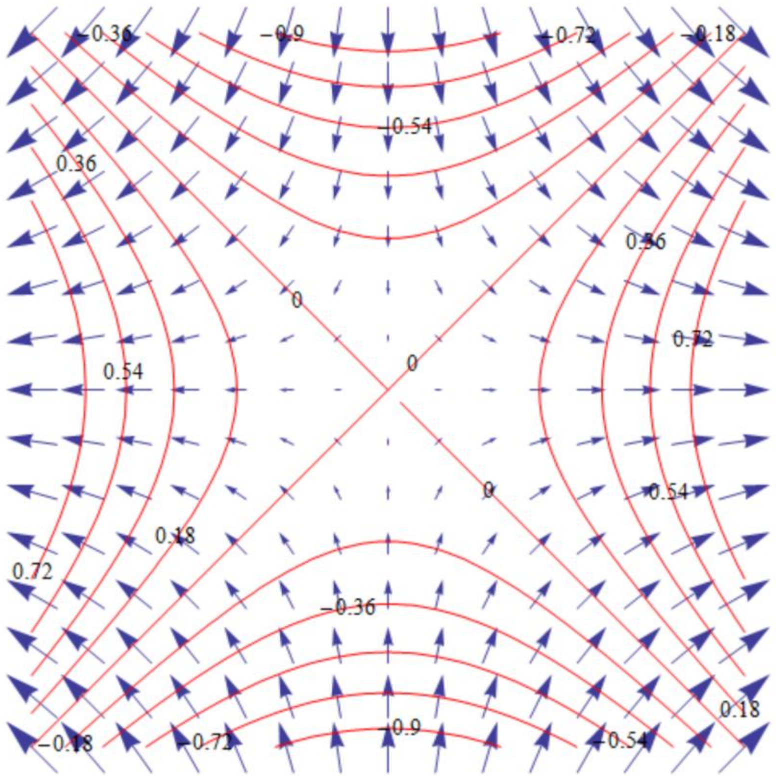


CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES: 2-D

- $f(\boldsymbol{x}) = x_1^2 + x_1 x_2 + 2x_2^2, \boldsymbol{\nabla} f = \begin{pmatrix} 2x_1 + x_2 \\ x_1 + 4x_2 \end{pmatrix}$
- Calcular el Hessiano, vectorizar f(**x**), e intuir el signo de los autovalores.

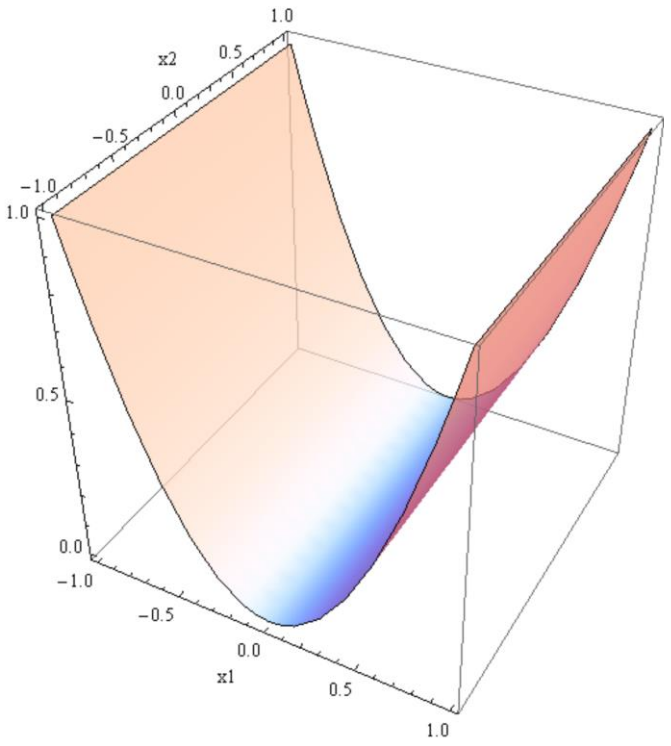


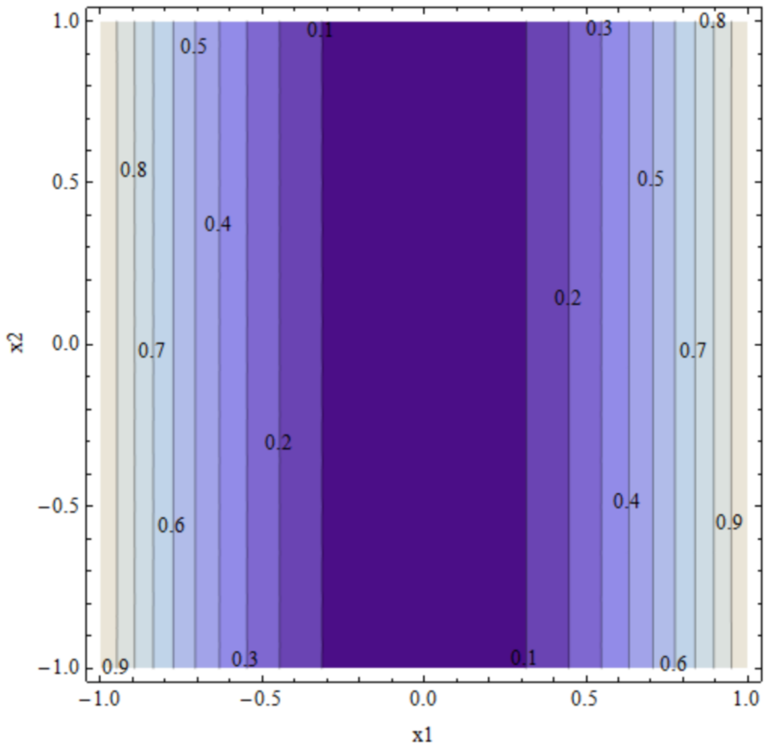


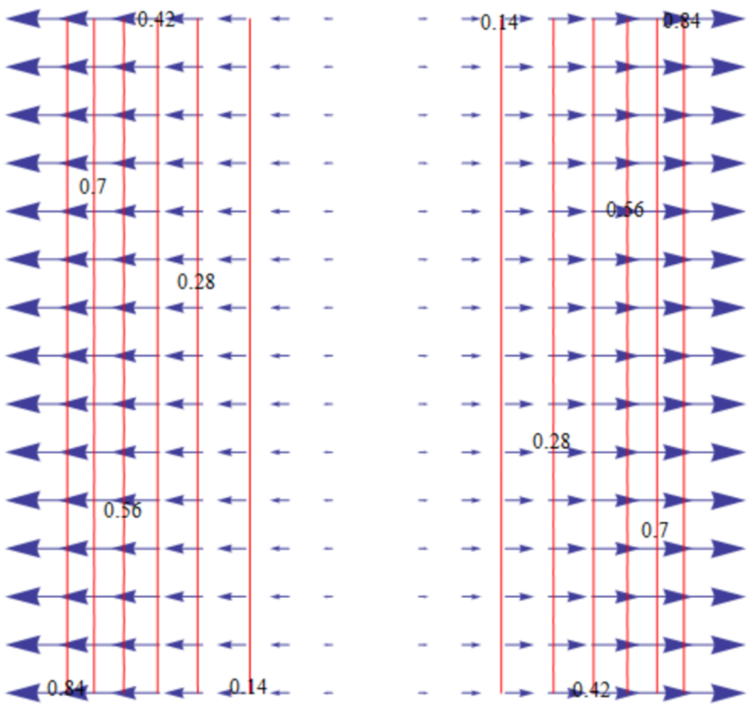


CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES: 2-D

- $f(\boldsymbol{x}) = x_1^2 - x_2^2, \nabla f = \begin{pmatrix} 2x_1 \\ -2x_2 \end{pmatrix}$
- Calcular el Hessiano, vectorizar f(**x**), e intuir el signo de los autovalores.

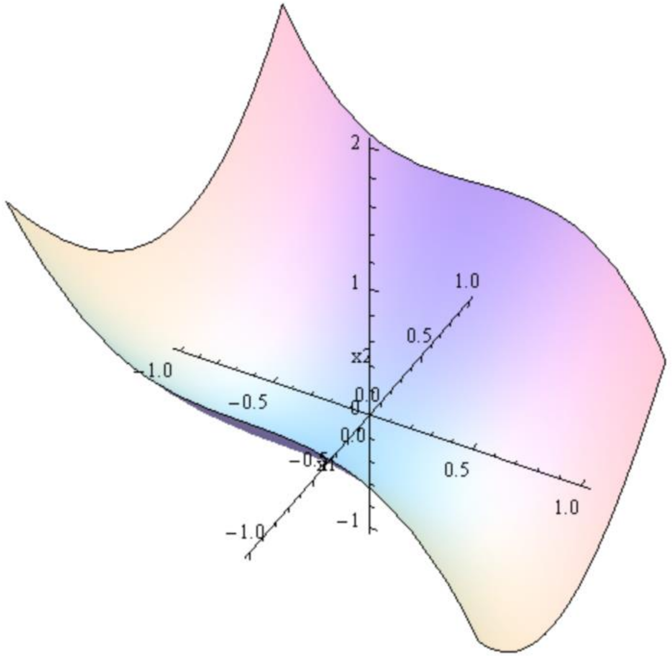


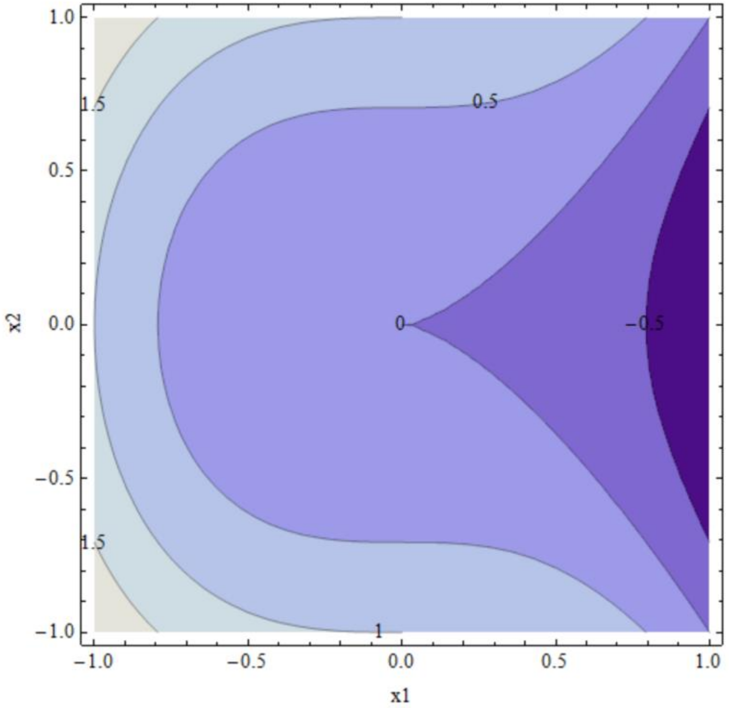


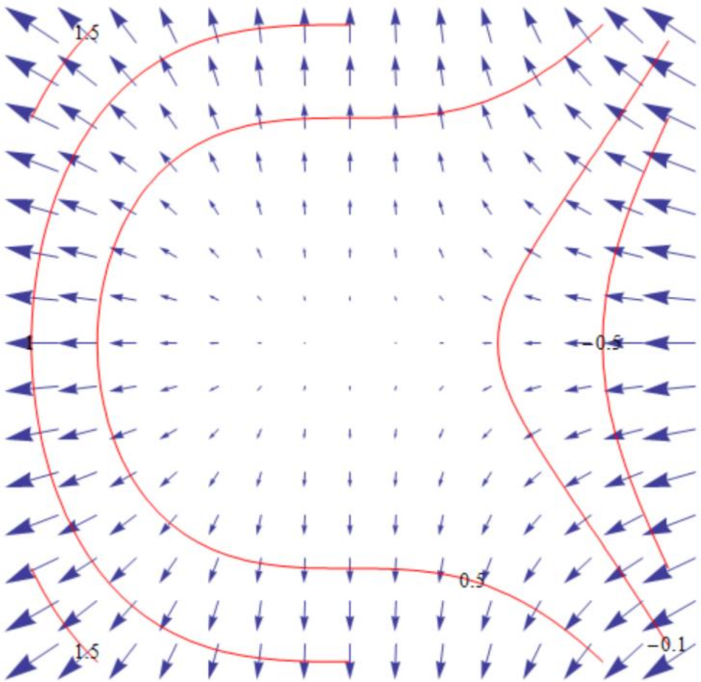


CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES: 2-D

- $f(\boldsymbol{x}) = x_1^2, \boldsymbol{\nabla} f = \begin{pmatrix} 2x_1 \\ 0 \end{pmatrix}$
- Calcular el Hessiano, vectorizar f(**x**), e intuir el signo de los autovalores.







CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES: 2-D

- Puede ocurrir que en alguna dirección las primera y segunda derivadas sean cero pero la tercera no. En ese caso, en esa dirección, ocurre el muy conocido caso en que un punto tiene derivada cero pero no es ni mínimo ni máximo.
- $f(\boldsymbol{x}) = -x_1^3 + x_2^2, \boldsymbol{\nabla} f = \begin{pmatrix} -3x_1^2 \\ 2x_2 \end{pmatrix}$
- Calcular el Hessiano, vectorizar f(**x**), e intuir el signo de los autovalores.

CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES

- El gradiente es perpendicular en cada punto a la curva de nivel (o a la superficie, o hiper superficie de nivel, dependiendo del numero de variables de nuestro problema) que pasa por dicho punto.
- Recordemos la serie de Taylor a primer orden: $f(x_1, \ldots, x_d) = f(a_1, \ldots, a_d) + \sum_{j=1}^{d} \frac{\partial f(a_1, \ldots, a_d)}{\partial x_j} (x_j - a_j)$
- Si la función es suave y el punto **x** está:
  1. A una distancia "minúscula" de **a** ($|d\boldsymbol{x}| = |\boldsymbol{x} - \boldsymbol{a}| < \epsilon$)
  2. Sobre la misma curva de nivel que **a** de modo que $f(\boldsymbol{x}) = f(\boldsymbol{a})$.
- Entonces el vector $d\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{a}$ es tangente a la curva de nivel (o a la superficie, o hiper superficie de nivel) y, escrita la ecuación de Taylor en forma vectorial: $\nabla f \cdot d\boldsymbol{x} = 0$

  significa que el gradiente es ortogonal a la tangente de la curva de nivel.
  - https://www.geogebra.org/m/iTZ19OCj
- Esto vale en cualquier dimensión.
  - Ver https://ucema.edu.ar/publicaciones/download/documentos/679.pdf y https://ucema.edu.ar/publicaciones/download/documentos/680.pdf

CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES

- El campo vectorial gradiente de una función tiene las siguientes propiedades:

1. Es perpendicular (ortogonal) a las curvas de nivel en cada punto.
2. La dirección del gradiente es la dirección donde la función aumenta más. Es la dirección del ascenso más empinado.
3. La magnitud del gradiente es proporcional a la tasa de crecimiento de la función en cada punto.
4. En particular, en un máximo local (o mínimo), donde la tasa de crecimiento es cero, el gradiente es cero.

CURVAS DE NIVEL, CAMPOS VECTORIALES Y GRADIENTES

- Explorar el software del capitulo 6 del del libro de Robert Johansson - Numerical Python_ Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib-Apress (2019).

HW 4

- Vamos a programar un optimizador (hacerlo en Python):

1. 1-D:
   a. Dado $f(x)$, queremos crear una función en Python para aproximar el mínimo (local).
   b. Elegimos un valor pequeño de Δx para calcular derivadas, una constante α que va a determinar el tamaño del paso, y otra constante pequeña ε que va a determinar cuándo paramos.
   c. Creamos dos listas vacías: X y F que vamos a ir llenando.
   d. Empezamos en un punto arbitrario $x_i, i = 0$, evaluamos $f(x_i)$. Hacemos $X[0] = x_0$ y $F[0] = f(x_0)$
   e. Calculamos numéricamente la derivada en ese punto: $f'(x_i) = \frac{f(x_i+\Delta x)-f(x_i)}{\Delta x}$
   f. Calculamos $x_{i+1} = x_i - \alpha \cdot f'(x_i)$ (por que el signo menos?) y $f(x_{i+1})$. Hacemos $X[i+1] = x_{i+1}$ y $F[i+1] = f(x_{i+1})$
   g. Verificamos si $|f(x_{i+1}) - f(x_i)| < \varepsilon$. Si se cumple la condición, frenamos, sino volvemos al paso 4 con $i = i + 1$ y repetimos hasta que la condición se verifique (como para que no ocurra que nunca pare si la función no tuviera mínimo, podemos además forzar que si $i = i_{max}$ para algún i_{max}, el programa termina).
   h. Los outputs de la función deben los valores i_{ultimo}, y las listas X y F. Los valores $X[i_{ultimo}] = x_{ultimo}$ y $F[i_{ultimo}] = f(x_{ultimo})$ corresponden a una ε-aproximación al mínimo local buscado. Pero además tenemos toda la historia del camino y los valores de la función que la misma fue tomando. Graficar estos valores en un diagrama (x, f).
   i. Aplicar el optimizador que acaban de programar a las funciones $f(x) = x - x^2 + x^4$ y $g(x) = x - 3x^2 + x^4$. Ver esas funciones en el Geogebra colgado en el webcampus y elegir estratégicamente varios puntos de partida x_0. Explicar por qué en el caso de f, si uno elige α lo suficientemente pequeña, siempre converge al mismo valor y en el caso de g no. Comprobar que si α es lo suficientemente grande el algoritmo NO converge, explicar por qué.

HW 4

- Vamos a programar un optimizador (hacerlo en Python):

2. 2-D:

   a. Dado $f(x)$, donde $\boldsymbol{x} = \binom{x_0}{y_0} \in \mathbb{R}^2$, queremos aproximar el mínimo (local).

   b. Elegimos un valor pequeño de $\Delta x = \Delta y$ para calcular derivadas, una constante α que va a determinar el tamaño del paso, y otra constante pequeña ε que va a determinar cuándo paramos.

   c. Creamos tres listas vacías: X, Y, y F que vamos a ir llenando.

   d. Empezamos en un punto arbitrario $\binom{x_0}{y_0}$, $x_i, i = 0, , y_i, i = 0$, evaluamos $f(x_i, y_i)$. Hacemos $X[0] = x_0$, $\mathrm{Y}[0] = y_0$ y $F[0] = f(x_0, y_0)$.

   e. Calculamos numéricamente las componentes del gradiente en ese punto: $\nabla_x f(x_i, y_i) = \frac{f(x_i+\Delta x, y_i) - f(x_i, y_i)}{\Delta x}$, $\nabla_y f(x_i, y_i) = \frac{f(x_i, y_i+\Delta y) - f(x_i, y_i)}{\Delta y}$.

   f. Calculamos $x_{i+1} = x_i - \alpha \cdot \nabla_x f(x_i, y_i)$, $y_{i+1} = y_i - \alpha \cdot \nabla_y f(x_i, y_i)$ y $f(x_{i+1}, y_{i+1})$. Hacemos $X[i+1] = x_{i+1}$, $\mathrm{Y}[i+1] = y_{i+1}$ y $F[i+1] = f(x_{i+1}, y_{i+1})$

   g. Verificamos si $|f(x_{i+1}, y_{i+1}) - f(x_i, y_i)| < \varepsilon$. Si se cumple la condición, frenamos, sino volvemos al paso 4 con $i = i + 1$ y repetimos hasta que la condición se verifique (como para que no ocurra que nunca pare si la función no tuviera mínimo, podemos además forzar que si $i = i_{max}$ para algún i_{max}, el programa termina).

   h. Los outputs de la función deben los valores i_{ultimo}, y las listas X, Y y F. Los valores $X[i_{ultimo}] = x_{ultimo}$, $\mathrm{Y}[i_{ultimo}] = y_{ultimo}$ y $F[i_{ultimo}] = f(x_{ultimo})$ corresponden a una ε-aproximación al mínimo local buscado. Pero además tenemos toda la historia del camino y los valores de la función que la misma fue tomando.

   i. Graficar en un mismo grafico en el plano (x, y), para las funciones $f(x, y) = (x$-1)**4 $+$ $(y$-1)**4 - $(x$-1)**2 - $(y$-1)**2 $+\ x\ + y$, y $g(x, y) = (x-1) ** 4\ +\ (y-1) ** 4\ - 2(x-1) ** 2\ - 3(y-1) ** 2\ +\ x\ + y$ (ver esas funciones en el JN y en el Geogebra colgados en el webcampus):

      i. Las curvas de nivel.

      ii. El camino $(X[i], Y[i])$ que hizo la maquina para llegar el mínimo. Notar que este camino (si α es lo suficientemente chico) es siempre perpendicular a las curvas de nivel.

      iii. En función de los gráficos de dichas funciones, elegir estratégicamente puntos iniciales y explicar por qué $f(x, y)$ siempre converge al mismo punto mientras que g(x, y) no.

      iv. Comprobar que si α es lo suficientemente grande el algoritmo NO converge, explicar por qué.