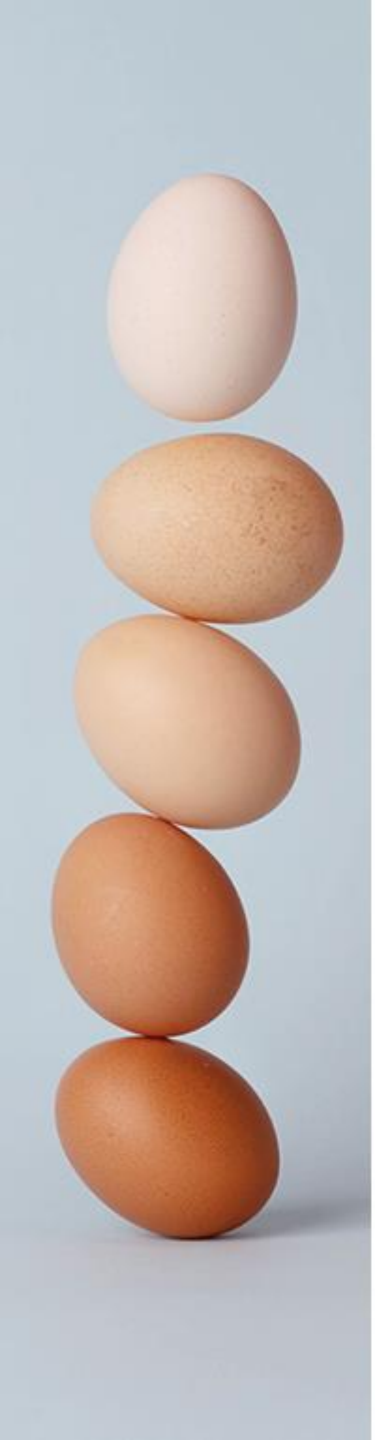
LA HUEVERA PERFECTA

Antón Riveiro Alonso, Laura Silva Loureiro
Cálculo y Análisis Numérico
Grupo Miércoles
{anton.riveiro, laura.silva.loureiro}@rai.usc.es

ÍNDICE:



1. Introducción al problema.

Una empresa dedicada en parte a la fabricación de envases para huevos nos ha encargado que obtengamos las coordenadas que se le deben pasar a las máquinas, encargadas del modelado de la base, donde se deberían situar los huecos para que encajen los huevos. Se trata de un envase con capacidad para seis huevos.

Especificaciones:

- El modelado sigue la forma de la función:

$$h(x, y) = 2 * (\cos(x) - \sin(y))$$

Es una función trigonométrica de dos variables: la x será la componente horizontal del plano donde está representada la función, mientras que la y es la componente vertical.

- Se busca un método para conseguir los máximos y mínimos de la función, que corresponderán con los huecos y del envase

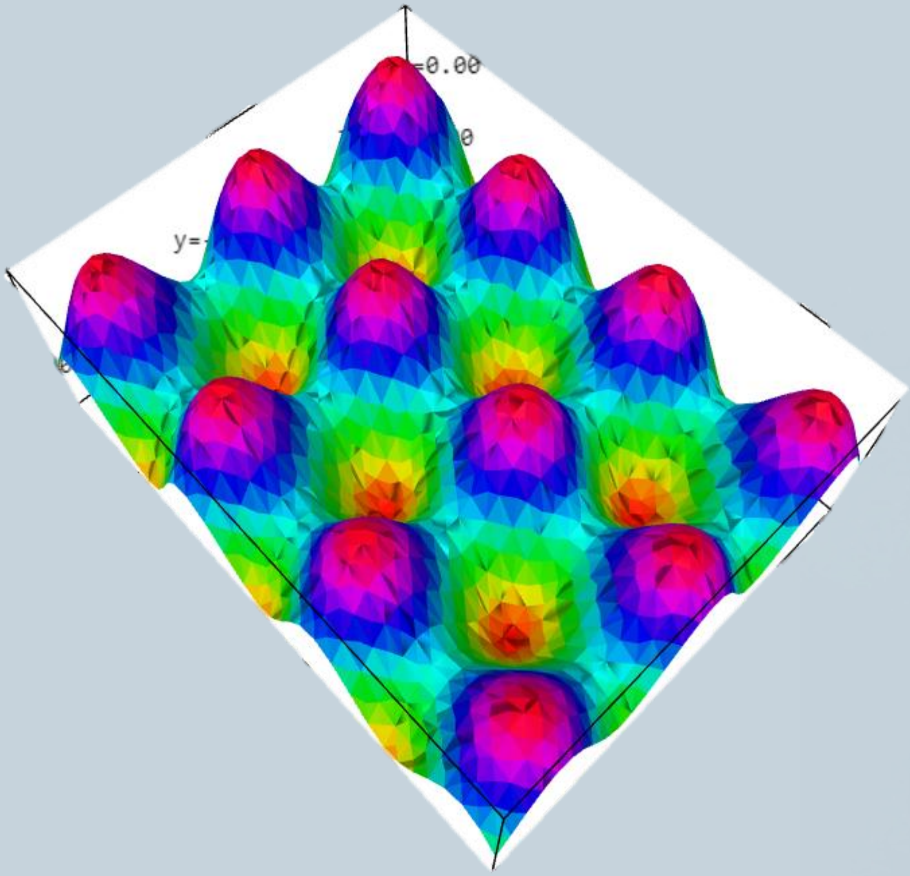
- Para el estudio del programa se utilizó la plataforma de computación en la nube de Cocalc [1].

Representación:

Para una mayor idea de la función con la que se va a trabajar, se representó de manera gráfica.

- Comandos:

1. `f(x,y)=2*(cos(x)-sin(y))`
2. `plot3d(f,(x,-14,8),(y,-10,7), adaptive = True)`

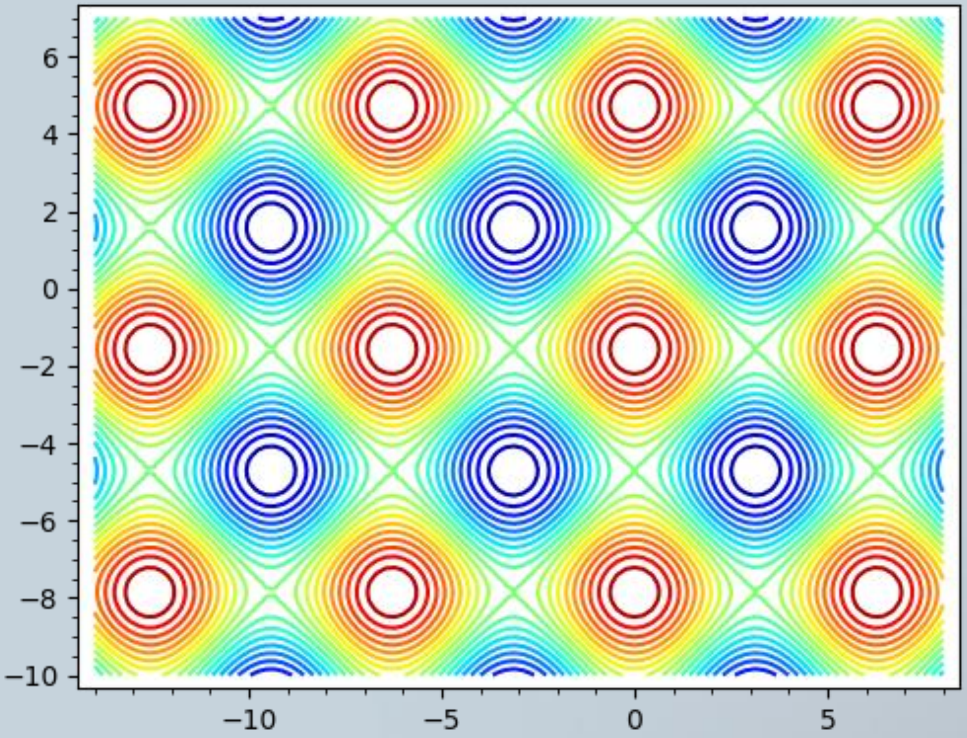


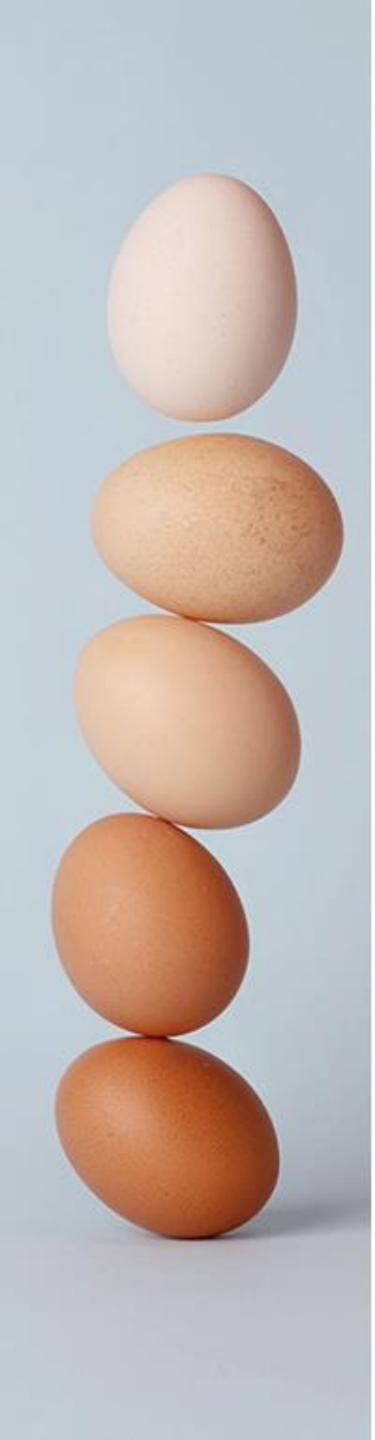
- También se realizó el mapeado de nivel para los intervalos x = (-14,8) e y = (-10,7), los dos medidos en radianes, que representa a la función h observada desde arriba. Las coordenada de los mínimos (huecos) y máximos se tomarán en base a esta gráfica.

- Comandos para hacer el mapeado:

```
1.  P = contour_plot(f,(x,-14,8),(y,-10,7),
    fill=False,cmap='jet',contours=20)
```

- <u>Máximos</u>: curvas rojas
- <u>Mínimos</u>: curvas azules
- <u>Puntos de silla</u>: cruces



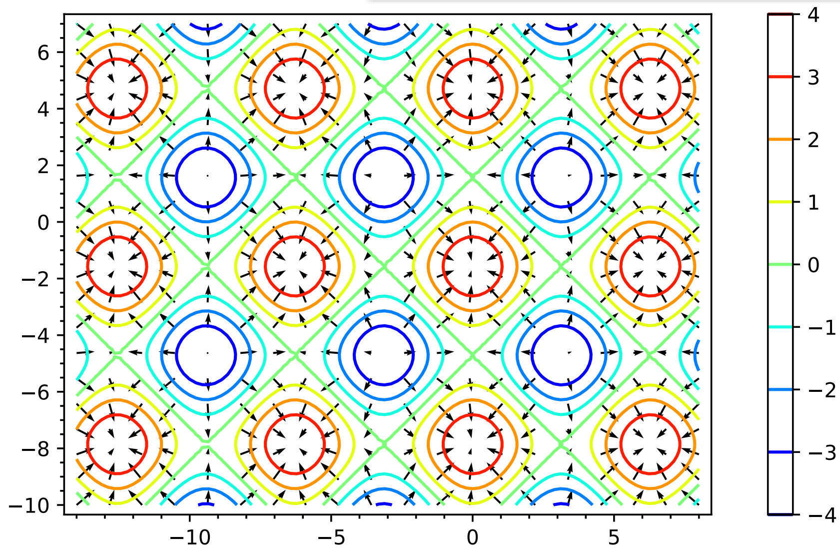


2. Cálculo de las derivadas parciales.

Para averiguar donde se encuentran los máximos y los mínimos de la función es necesario calcular su vector gradiente, formado por las derivadas parciales de $h(x,y)$.

Definimos el vector gradiente como:

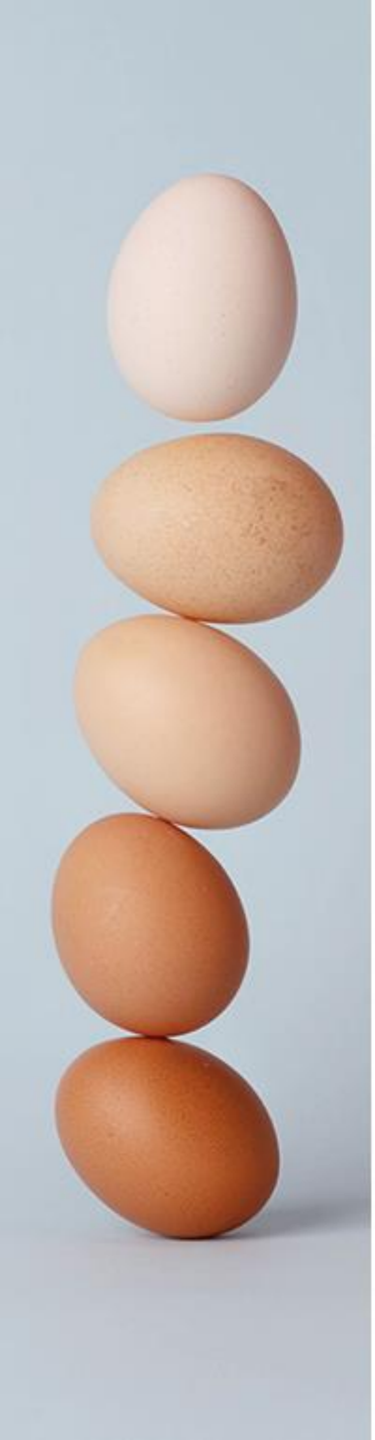
$$\nabla h(x,y) \; = \; (h_x(x,y), h_y(x,y))$$




```
Comandos sage:
2. Q = plot_vector_field(diff(f),
(x,-14,8), (y, -10,7))

3. P + Q
```

Los puntos críticos del vector gradiente, los puntos donde se anula $\nabla h(x,y) = 0$, que serán posibles máximos, mínimos o puntos de silla.



Se tienen que calcular por lo tanto las raíces del vector gradiente definido como: $f(x, y) = \nabla h(x, y)$

Para ello hay que resolver el siguiente sistema:

$$\begin{cases} f_{1}(x, y) = h_{x}(x, y) = 0 \\ f_{2}(x, y) = h_{y}(x, y) = 0 \end{cases}$$

Las derivadas parciales de nuestra función $h(x, y) = 2 * (\cos(x) - \sin(y))$ da un sistema no lineal:

$$\begin{cases} f_{1}(x, y) = -2 * \sin(x) \\ f_{2}(x, y) = -2 * \cos(y) \end{cases}$$

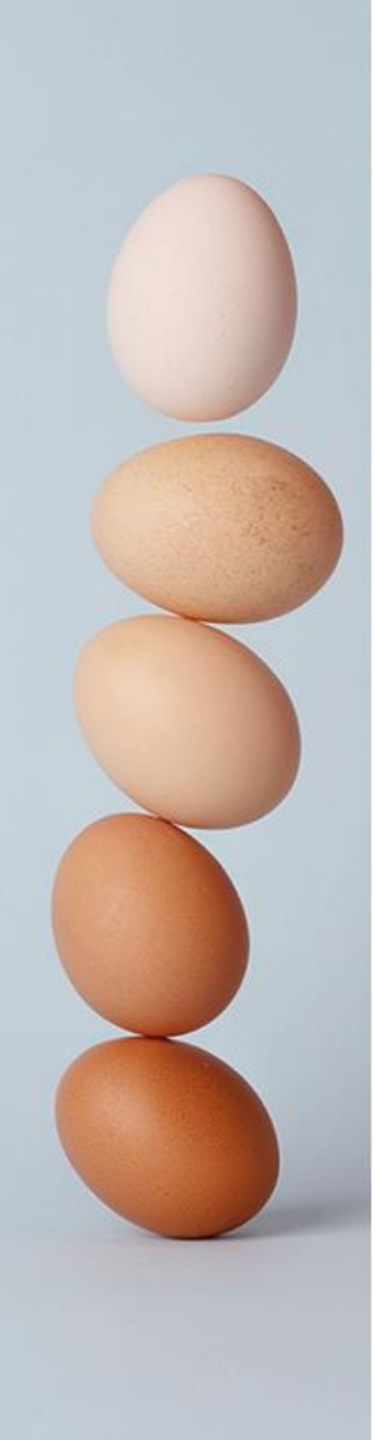
Las derivadas parciales de nuestro vector gradiente son:

$$f_{1x}(x, y) = h_{xx}(x, y) = -2 * \cos(x)$$

$$f_{2x}(x, y) = h_{yx}(x, y) = 0$$

$$f_{1y}(x, y) = h_{xy}(x, y) = 0$$

$$f_{2y}(x, y) = h_{yy}(x, y) = 2 * \sin(y)$$

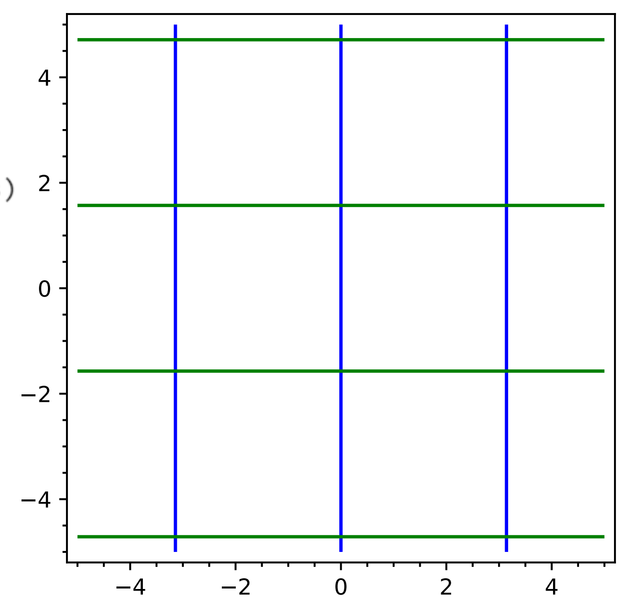


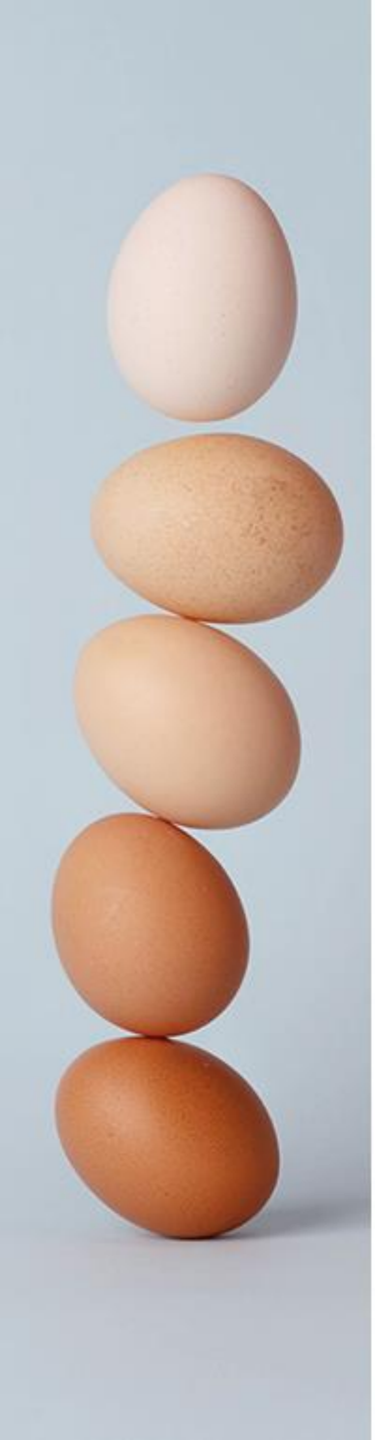
3. Matriz jacobiana y matriz hessiana.

Calcular la matriz Jacobiana de la función $f(x,y) = \nabla h(x,y)$ equivale a calcular la matriz Hessiana $(Hh(x,y))$ de la función $h(x,y)$:

$$Df(x,y) = Hh(x,y) = \begin{pmatrix} -2\cdot\cos(x) & 0 \\ 0 & 2\cdot\sin(y) \end{pmatrix}$$

```
def pinta_raiz(f1,f2):
    args = [(x,-5,5), (y, -5,5)]
    kwds = {'contours':[0], 'fill':False}
    P = contour_plot(f1,*args, cmap=['blue'], **kwds)
    Q = contour_plot(f2,*args, cmap=['green'], **kwds)
    show(P + Q)
```





4. Método de Newton.

Para aplicar este método no basta con obtener las derivadas parciales de nuestra función, si no que también la matriz Jacobiana ($Df(x, y)$), conformada por las derivadas parciales del vector gradiente.

Una vez obtenidos el vector gradiente y la matriz Jacobiana, los comandos en Cocalc[1] para obtener una de las raíces del vector gradiente (posibles máximos y mínimos buscados) son:

Comandos:

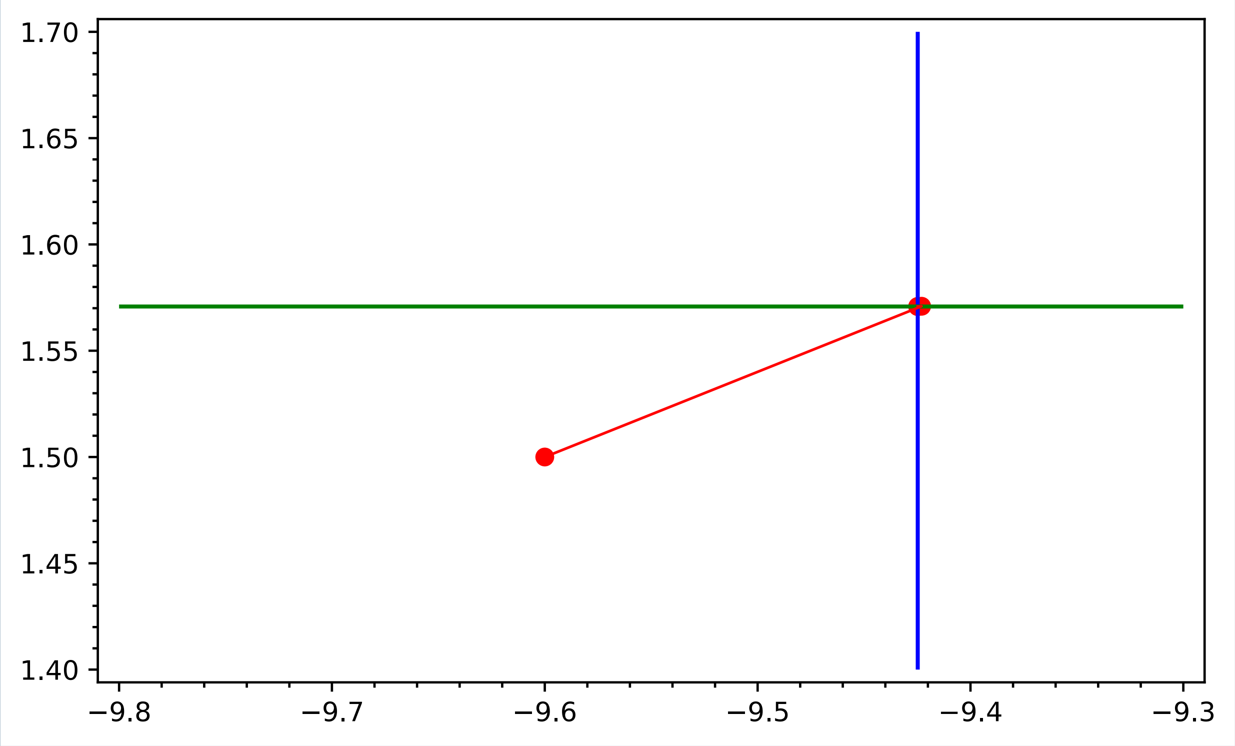
```
1. load('resolvesis.sage’)
2. x0 = newton(diff(f),diff(f,2),(-9,2))[0]
3. i = newton(diff(f),diff(f,2),(-9,2))[1]
4. x0, i
Sage: ((-9.42477796076938, 1.5707963267948966), 4)
```

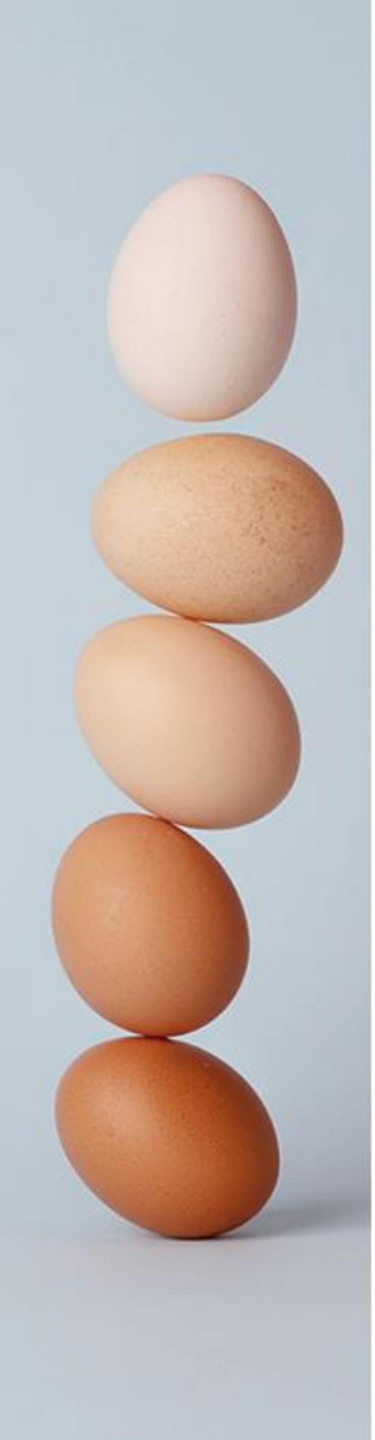
El primer miembro del vector(x0) es la raíz, y el segundo el numero de iteraciones (i).

Gráfica de Newton:

Comandos:

1. `load('resolvesis.sage’)`
2. `debuxa_newton(diff(f),diff(f,2),(-9.6,1.5),(-9.8,-9.3,1.4,1.7))`





5. Método del descenso.

Norma de la función como la suma de los cuadrados de los componentes:

$$G(\mathrm{x}, y) = ||\mathrm{f}(\mathrm{x},\mathrm{y})||^2 = (f_1(\mathrm{x},\mathrm{y})^2 + f_2(\mathrm{x},\mathrm{y})^2)$$

$$G(x, y) = 4 \cdot \cos(y)^2 + 4 \cdot \sin(x)^{\ 2}$$

$$\nabla G(x, y) = (2 \cdot f_1(x, y) + 2 \cdot f_2(x, y)) \cdot Df(x, y)$$

$$\nabla G(x, y) = (G_x(x, y)\ ,\ G_y(x, y))$$

$$\nabla G(x, y) = (8 \cdot \cos(\mathrm{x}) \cdot \sin(\mathrm{x}),\ -8 \cdot \cos(\mathrm{y}) \cdot \sin(\mathrm{y}))$$

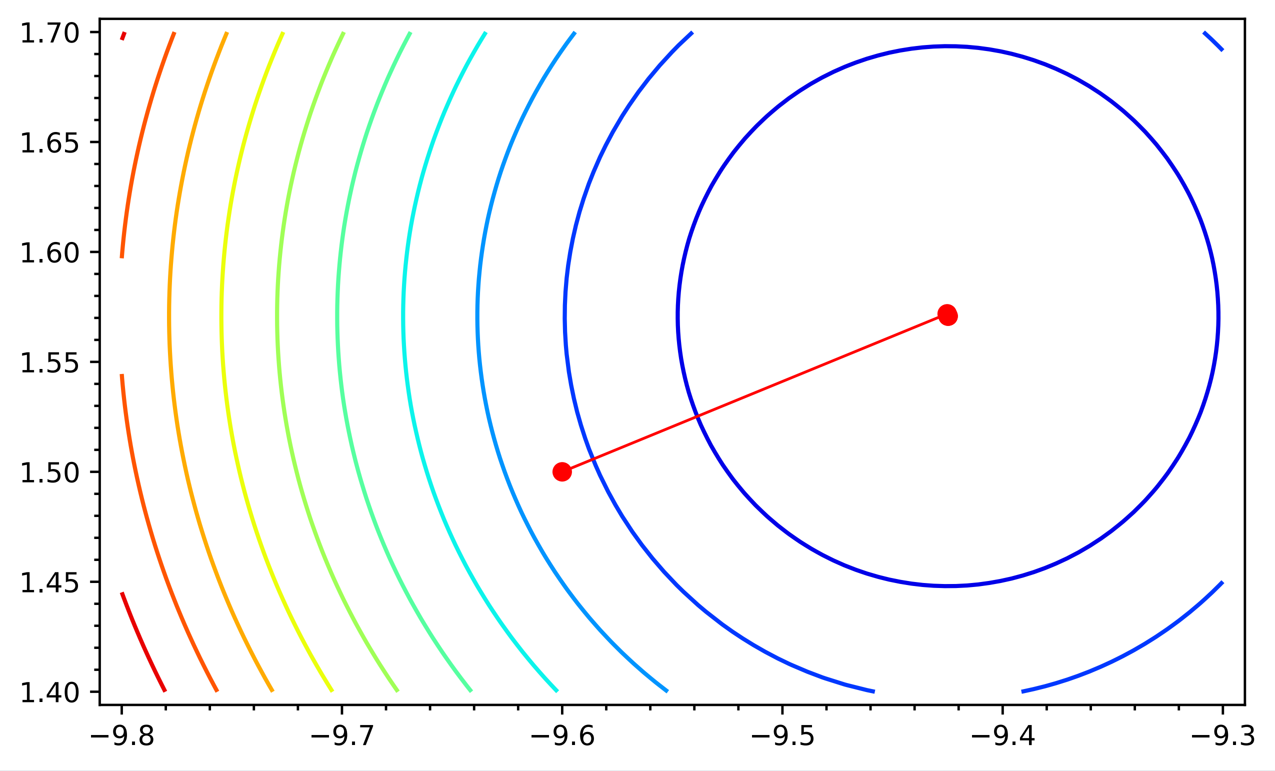
Comandos Sage:

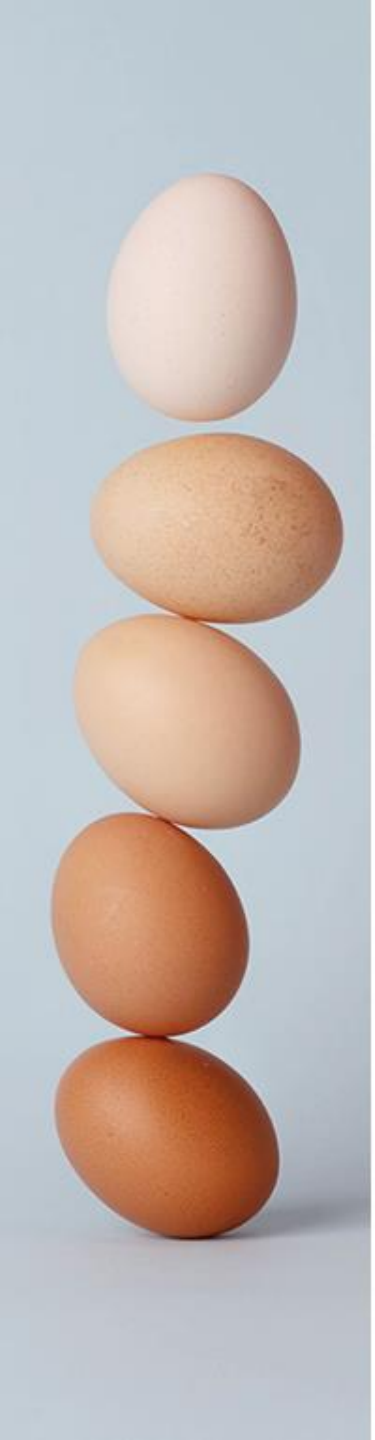
```
1. load('minimiza_Sage9.sage’)
2. descenso(g, diff(g), (-9,2))[0]
Sage: Gradiente nulo, posible minimo.
      ((-9.42477960796272, 1.5707963267985483), 3)
```

Gráfica del descenso

```
debuxa_descenso(G,diff(G),(-9.6,1.5),(-9.8,-9.3,1.4,1.7))
```

```
def debuxa_descenso(f, df, x0, d):
    r,k,v = descenso(f, df, x0)
    args = [(x,d[0],d[1]), (y,d[2],d[3])]
    kwds = {'contours': 10, 'fill': False, 'cmap': 'jet'}
    h1 = contour_plot(f(x,y), *args, **kwds)
    h3 = point(v, marker='o', size=50, color='red')+line(v, color='red')
    show(h1+h3)
```





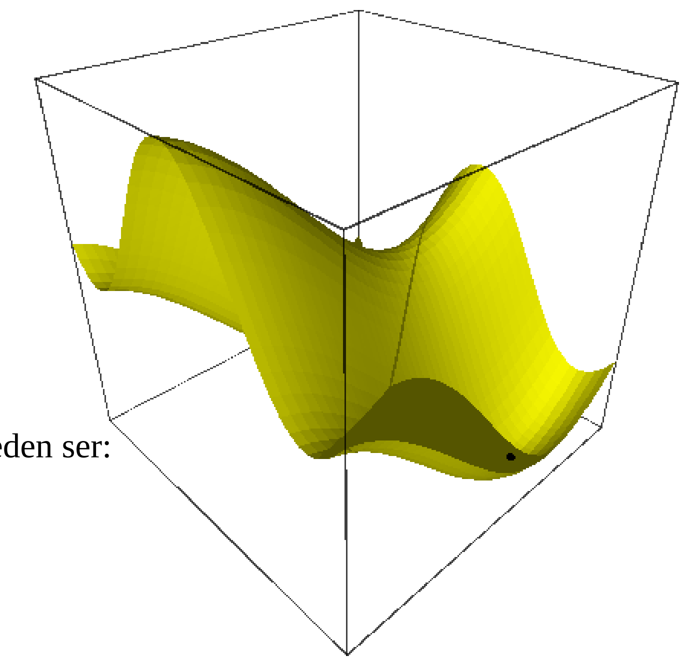
6. Estudio del criterio de la Hessiana.

Comandos:

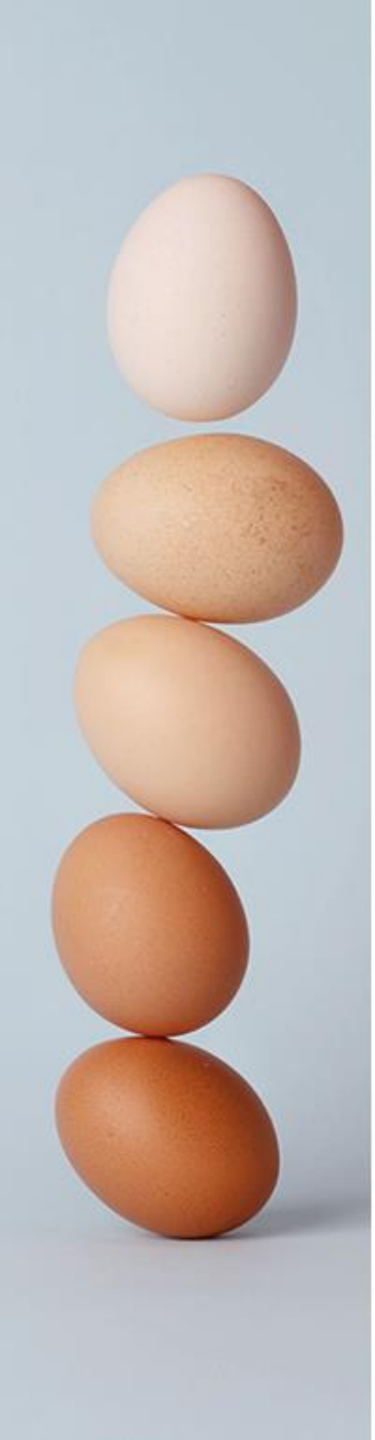
```
1. criterio_hessiana(G,-9.42477796076938, 1.5707963267948966)
```

```
Sage: H[0][0]=8.00000000000000, det(H)=64.0000000000000; o
punto e un minimo local.
```

```
P = plot3d(G,(x,-10,-8),(y,-2,2), viewer= 'tachyon',color= 'yellow’)
J = point ((-9.42477796076938, 1.5707963267948966,G(-9.42477796076938,
1.5707963267948966)),color='black’)
P+J
```



Según el criterio de la Hessiana nuestros puntos pueden ser:
- Un máximo si hessiano > 0 y h_{xx} <0
- Un mínimo si hessiano > 0 y h_{xx} >0
- Un punto de silla si hessiano < 0



7. Bibliografía

[1] Cocalc. [en línea], [sin fecha]. [Consulta: 18 April 2022]. Disponible en: https://cocalc.com/.

- SageMath documentation. [en línea], [sin fecha]. [Consulta: 18 April 2022]. Disponible en: https://doc.sagemath.org/.