

Homework N°4

Junghanss, Juan Cruz & Marco, Tomás Guido

Métodos Cuantitativos para Ciencias Sociales y Negocios - Prof. Sergio Pernice

April 19, 2021

- **Curvas de Nivel, Campos Vectoriales y Gradientes 2-D:**

Replicar los gráficos de curvas de nivel, los campos vectoriales y gradientes de cada forma cuadrática expresada en la presentación de clase N°5 (15/04 - MCCSN Tema 5).

Para el desarrollo, comenzaremos mostrando la resolución del ejercicio en Python y por último, mostraremos un anexo con la misma resolución en LaTeX.

```
[1]: import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
#%%matplotlib qt
%matplotlib inline
```

```

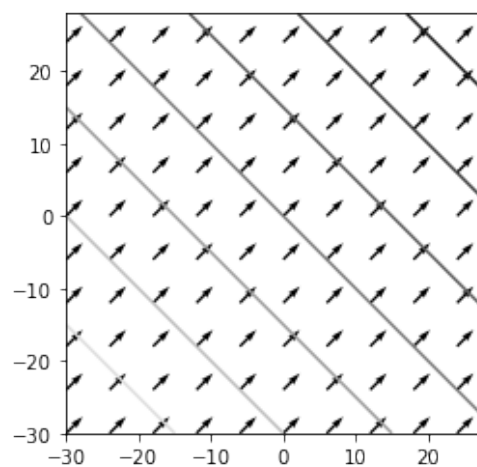
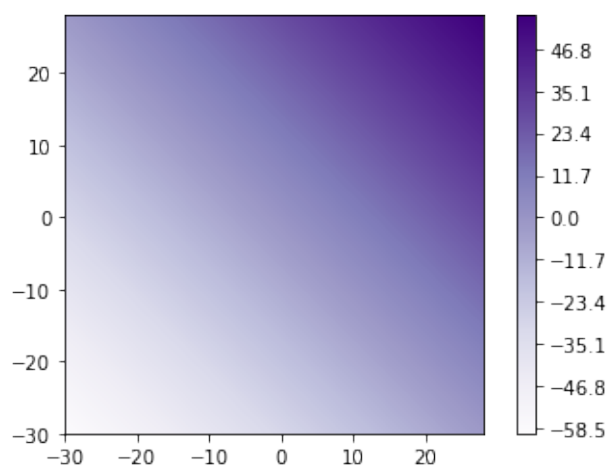
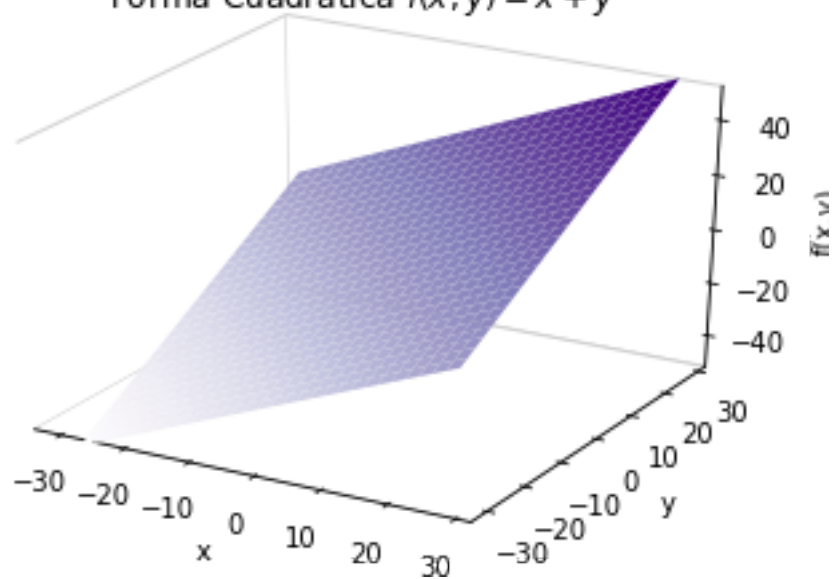
[2]: fig, ax = plt.subplots(1, 1)
feature_xy = np.arange(-30, 30, 2)
x, y = np.meshgrid(feature_xy, feature_xy)
z = x + y
ax.set_aspect(1)
c1 = ax.contour(x, y, z, 1000, cmap='Purples', alpha=1)
cbar = plt.colorbar(c1)
plt.show()

fig, ax = plt.subplots(1, 1)
c2 = ax.contour(x, y, z, 10, cmap='binary', alpha=1)
ax.set_aspect(1)
a,b = np.meshgrid(np.arange(-30, 30, 6), np.arange(-30, 30,6))
c = a + b
v, u = np.gradient(c, .2, .2)
q = ax.quiver(a,b,u,v)
plt.show()

plt.figure()
ax = plt.axes(projection='3d')
ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='Purples', edgecolor=None)
ax.set(xlabel='x', ylabel='y', zlabel='f(x,y)', title='Forma Cuadrática_
→$f(x,y)=x+y$')
ax.set_zlim3d([-50, 50])
ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.grid(False)
ax.xaxis.pane.set_edgecolor('#DODODO')
ax.yaxis.pane.set_edgecolor('#DODODO')
ax.zaxis.pane.set_edgecolor('#DODODO')
ax.xaxis.pane.set_alpha(1)
ax.yaxis.pane.set_alpha(1)
ax.zaxis.pane.set_alpha(1)
plt.show()

```

Forma Cuadrática $f(x, y) = x + y$



```

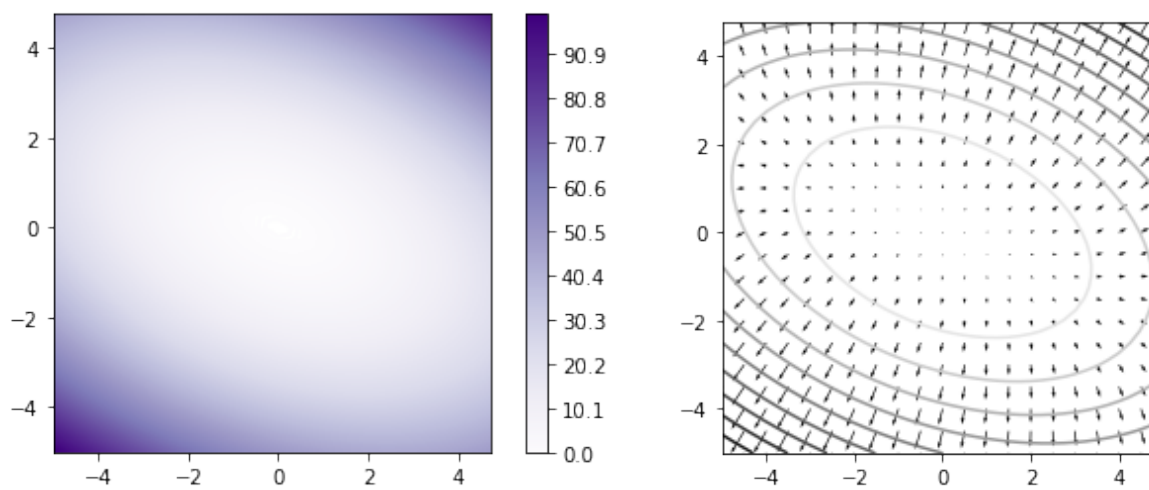
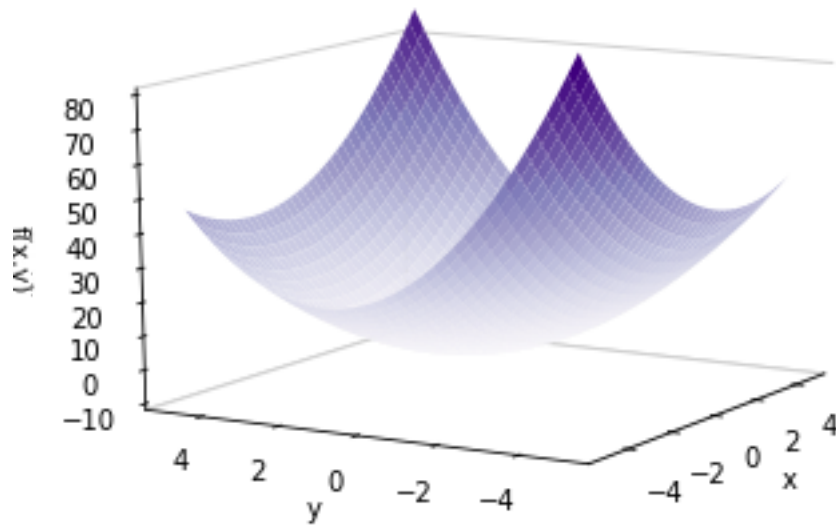
[3]: fig, ax = plt.subplots(1, 1)
feature_xy = np.arange(-5, 5, 0.25)
x, y = np.meshgrid(feature_xy, feature_xy)
z = x**2 + x*y + 2*y**2
ax.set_aspect(1)
c1 = ax.contour(x, y, z, 1000, cmap='Purples', alpha=1)
cbar = plt.colorbar(c1)
plt.show()

fig, ax = plt.subplots(1, 1)
c2 = ax.contour(x, y, z, 10, cmap='binary', alpha=1)
ax.set_aspect(1)
a,b = np.meshgrid(np.arange(-5, 5, .5), np.arange(-5, 5, .5))
c = a**2 + a*b + 2*b**2
v, u = np.gradient(c, .2, .2)
q = ax.quiver(a,b,u,v)
plt.show()

plt.figure()
ax = plt.axes(projection='3d')
ax.view_init(15,-150)
ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='Purples', edgecolor=None)
ax.set(xlabel='x', ylabel='y', zlabel='f(x,y)', title='Forma Cuadrática_
→$f(x,y)=x^2+x·y+2·y^2$')
ax.set_zlim3d([-10, 80])
ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.grid(False)
ax.xaxis.pane.set_edgecolor('#D0D0D0')
ax.yaxis.pane.set_edgecolor('#D0D0D0')
ax.zaxis.pane.set_edgecolor('#D0D0D0')
ax.xaxis.pane.set_alpha(1)
ax.yaxis.pane.set_alpha(1)
ax.zaxis.pane.set_alpha(1)
plt.show()

```

Forma Cuadrática $f(x, y) = x^2 + x \cdot y + 2 \cdot y^2$



```

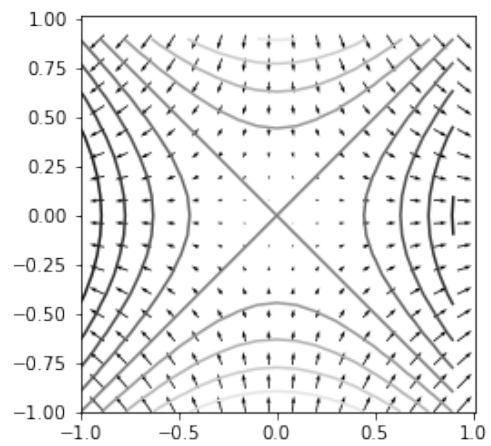
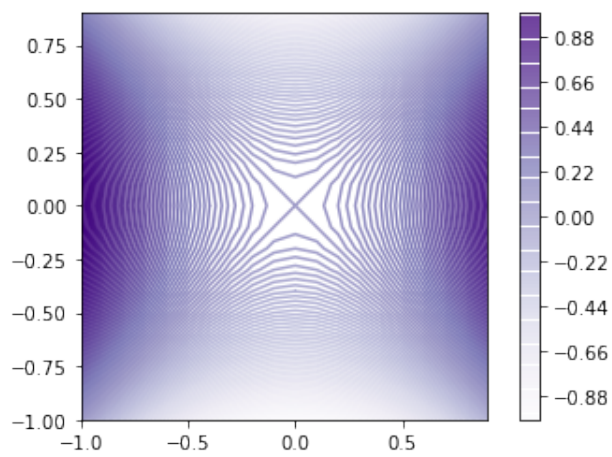
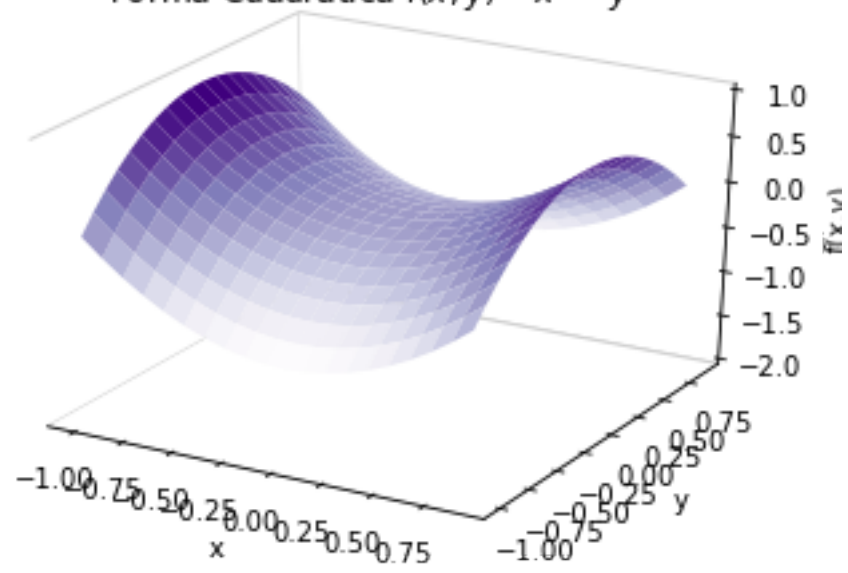
[4]: fig, ax = plt.subplots(1, 1)
feature_xy = np.arange(-1, 1, 0.10)
x, y = np.meshgrid(feature_xy, feature_xy)
z = x**2 - y**2
ax.set_aspect(1)
c1 = ax.contour(x, y, z, 100, cmap='Purples', alpha=1)
cbar = plt.colorbar(c1)
plt.show()

fig, ax = plt.subplots(1, 1)
c2 = ax.contour(x, y, z, 10, cmap='binary', alpha=1)
ax.set_aspect(1)
a,b = np.meshgrid(np.arange(-1, 1, .12), np.arange(-1, 1, .12))
c = a**2 - b**2
v, u = np.gradient(c, .5, .5)
q = ax.quiver(a,b,u,v)
plt.show()

plt.figure()
ax = plt.axes(projection='3d')
ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='Purples', edgecolor=None)
ax.set(xlabel='x', ylabel='y', zlabel='f(x,y)', title='Forma Cuadrática_
→$f(x,y)=x^2-y^2$')
ax.set_zlim3d([-2, 1])
ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
ax.grid(False)
ax.xaxis.pane.set_edgecolor('#DODODO')
ax.yaxis.pane.set_edgecolor('#DODODO')
ax.zaxis.pane.set_edgecolor('#DODODO')
ax.xaxis.pane.set_alpha(1)
ax.yaxis.pane.set_alpha(1)
ax.zaxis.pane.set_alpha(1)
plt.show()

```

Forma Cuadrática $f(x, y) = x^2 - y^2$



```

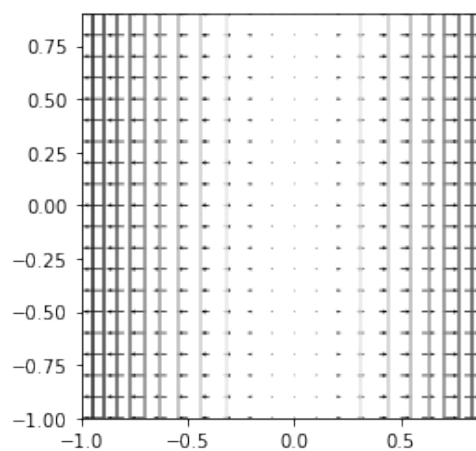
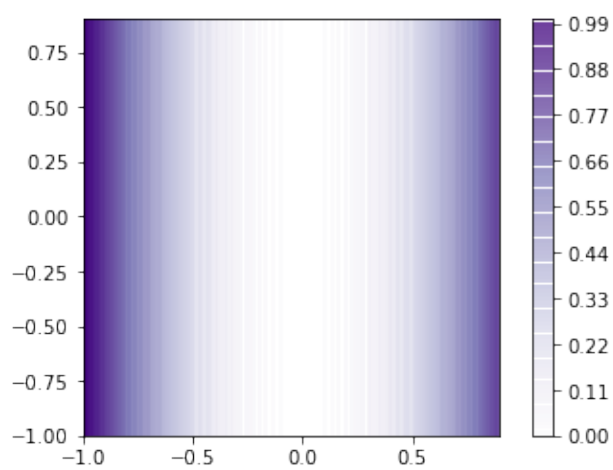
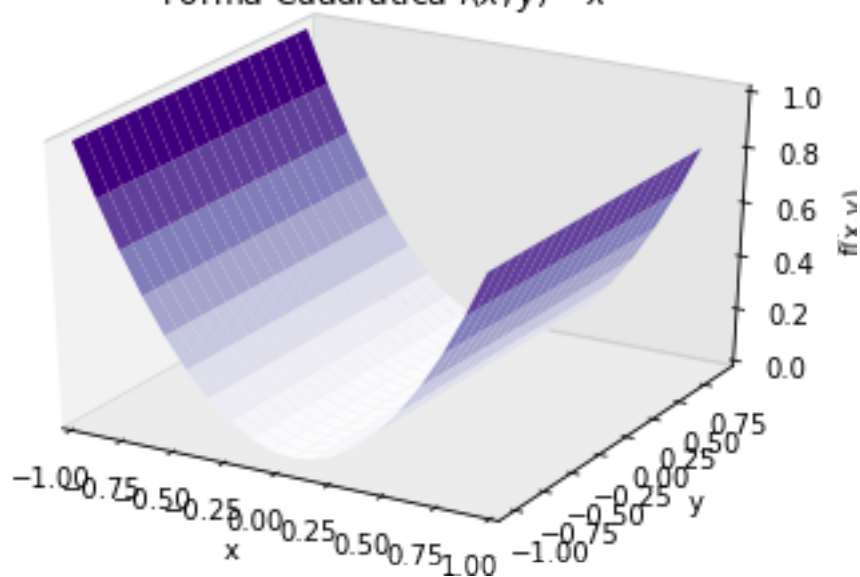
[5]: fig, ax = plt.subplots(1, 1)
feature_xy = np.arange(-1, 1, 0.10)
x, y = np.meshgrid(feature_xy, feature_xy)
z = x**2
ax.set_aspect(1)
c1 = ax.contour(x, y, z, 100, cmap='Purples', alpha=1)
cbar = plt.colorbar(c1)
plt.show()

fig, ax = plt.subplots(1, 1)
c2 = ax.contour(x, y, z, 10, cmap='binary', alpha=1)
ax.set_aspect(1)
a,b = np.meshgrid(np.arange(-1, 1, .1), np.arange(-1, 1, .1))
c = a**2
v, u = np.gradient(c, .5, .5)
q = ax.quiver(a,b,u,v)
plt.show()

plt.figure()
ax = plt.axes(projection='3d')
ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='Purples', edgecolor=None)
ax.set(xlabel='x', ylabel='y', zlabel='f(x,y)', title='Forma Cuadrática ↵
↪$f(x,y)=x^2$')
ax.set_xlim3d([-1,1])
ax.set_zlim3d([0, 1])
ax.grid(False)
ax.xaxis.pane.set_edgecolor('#D0D0D0')
ax.yaxis.pane.set_edgecolor('#D0D0D0')
ax.zaxis.pane.set_edgecolor('#D0D0D0')
ax.xaxis.pane.set_alpha(1)
ax.yaxis.pane.set_alpha(1)
ax.zaxis.pane.set_alpha(1)
plt.show()

```


Forma Cuadrática $f(x, y) = x^2$



```

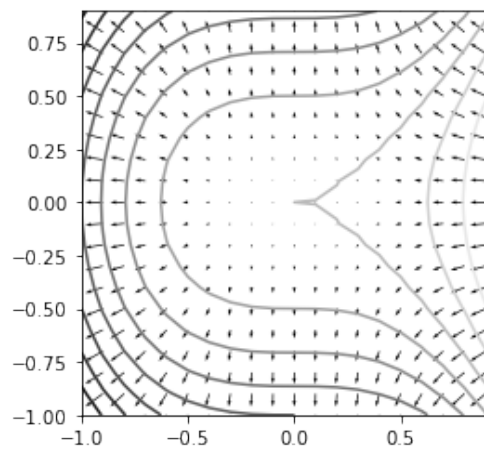
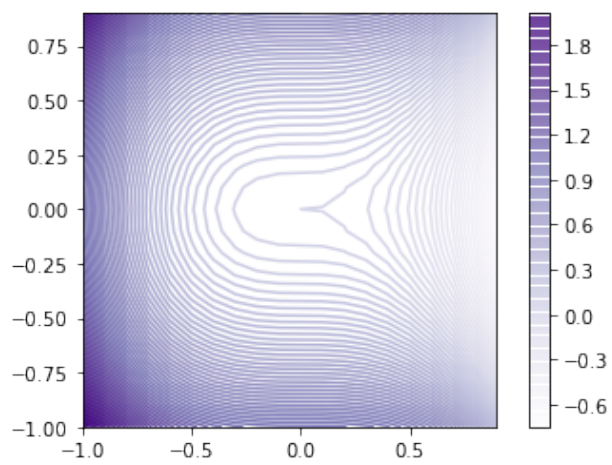
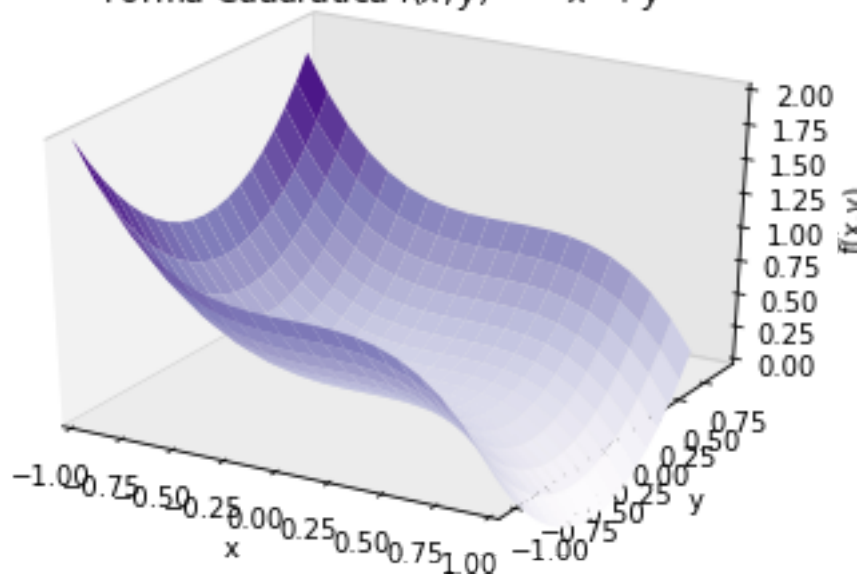
[6]: fig, ax = plt.subplots(1, 1)
feature_xy = np.arange(-1, 1, 0.10)
x, y = np.meshgrid(feature_xy, feature_xy)
z = -x**3 + y**2
ax.set_aspect(1)
c1 = ax.contour(x, y, z, 100, cmap='Purples', alpha=1)
cbar = plt.colorbar(c1)
plt.show()

fig, ax = plt.subplots(1, 1)
c2 = ax.contour(x, y, z, 10, cmap='binary', alpha=1)
ax.set_aspect(1)
a,b = np.meshgrid(np.arange(-1, 1, .1), np.arange(-1, 1, .1))
c = -a**3 + b**2
v, u = np.gradient(c, .5, .5)
q = ax.quiver(a,b,u,v)
plt.show()

plt.figure()
ax = plt.axes(projection='3d')
ax.plot_surface(x, y, z, rstride=1, cstride=1, cmap='Purples', edgecolor=None)
ax.set(xlabel='x', ylabel='y', zlabel='f(x,y)', title='Forma Cuadrática_
    ↳$f(x,y)=-x^3+y^2$')
ax.set_xlim3d([-1,1])
ax.set_zlim3d([0, 2])
ax.grid(False)
ax.xaxis.pane.set_edgecolor('#D0D0D0')
ax.yaxis.pane.set_edgecolor('#D0D0D0')
ax.zaxis.pane.set_edgecolor('#D0D0D0')
ax.xaxis.pane.set_alpha(1)
ax.yaxis.pane.set_alpha(1)
ax.zaxis.pane.set_alpha(1)
plt.show()

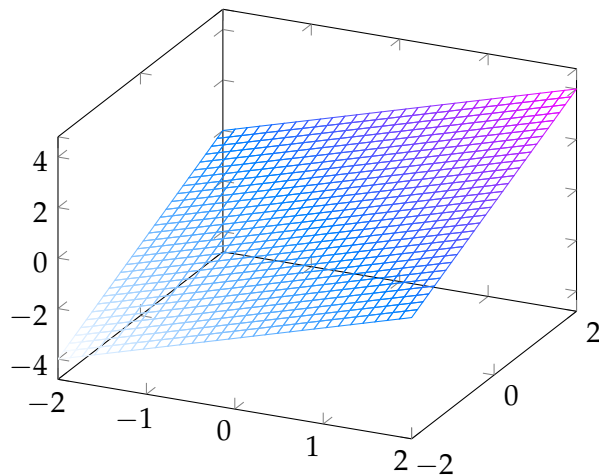
```

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

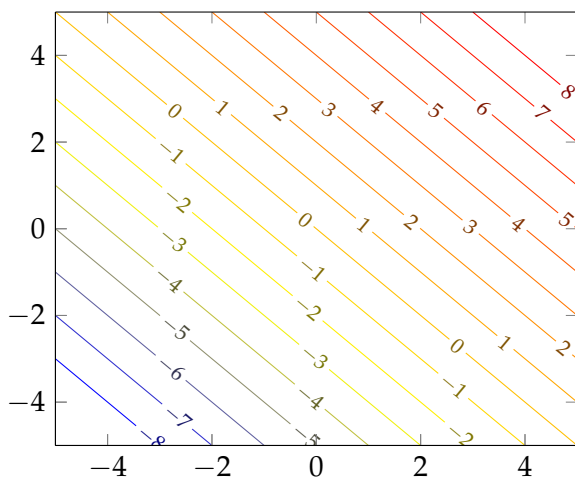


Anexo: gráficos hechos en LaTeX

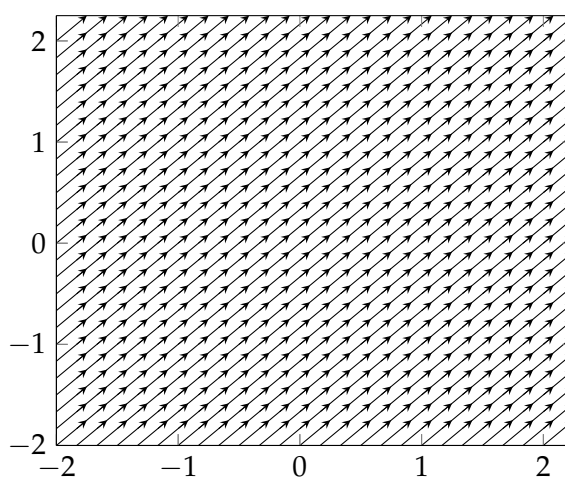
Forma cuadrática para $f(x, y) = x + y$



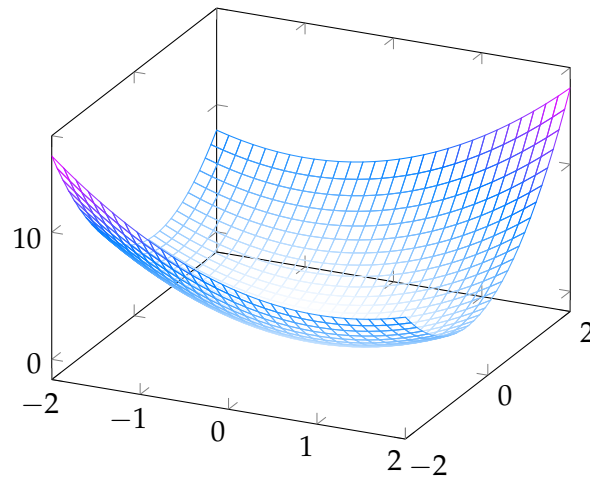
Curvas de nivel para $f(x, y) = x + y$



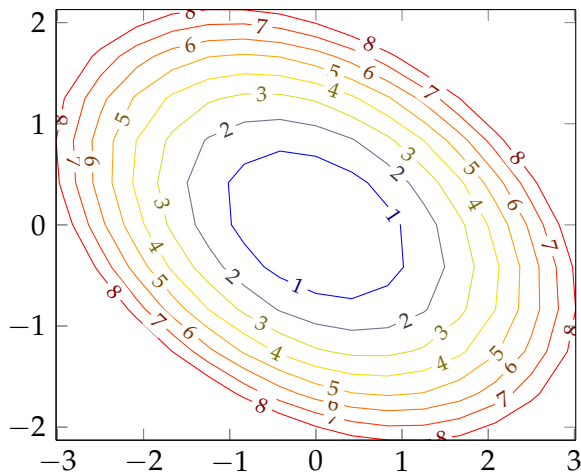
Vector gradiente para $f(x, y) = x + y$



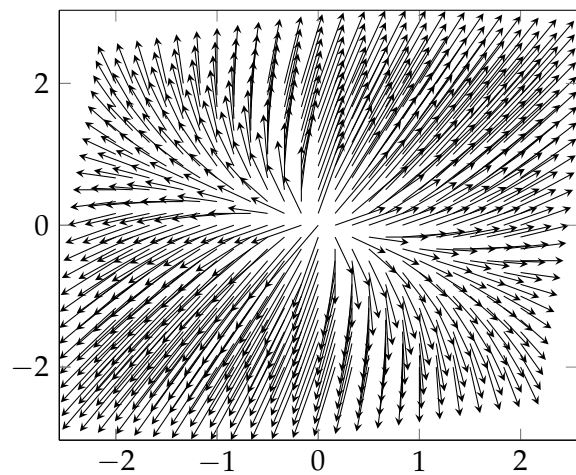
Forma cuadrática para $f(x, y) = x^2 + x \cdot y + 2 \cdot y^2$



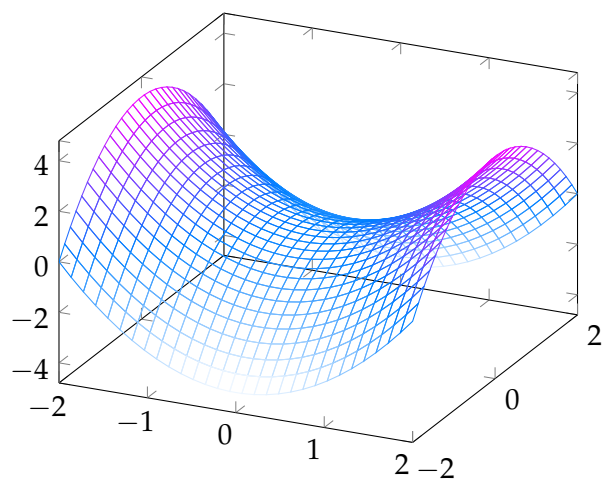
Curvas de nivel para $f(x, y) = x^2 + x \cdot y + 2 \cdot y^2$



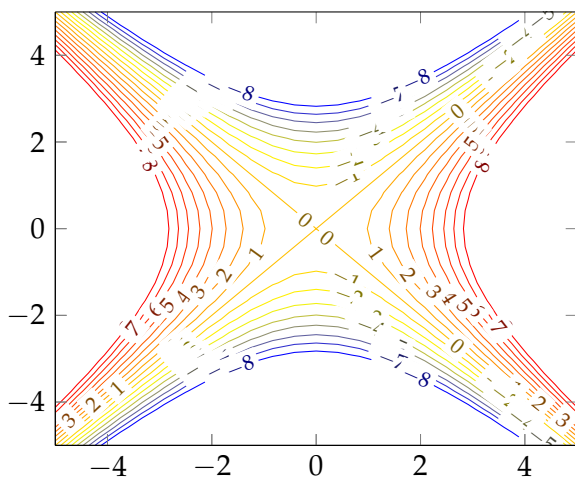
Vector gradiente



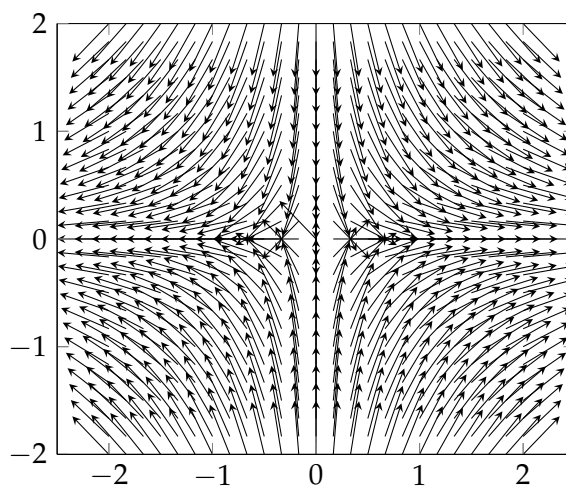
Forma cuadrática para $f(x, y) = x^2 - y^2$



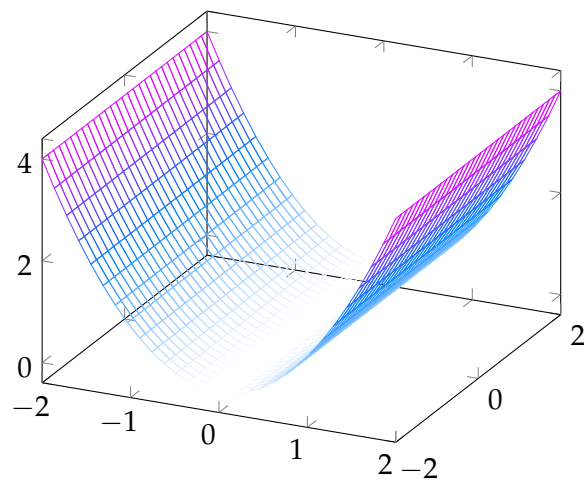
Curvas de nivel para $f(x, y) = x^2 - y^2$



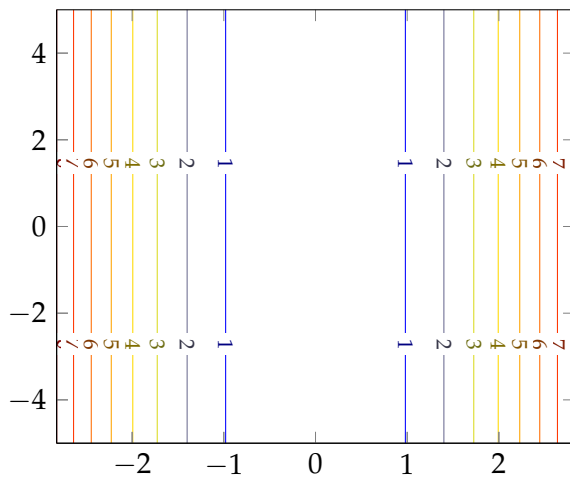
Vector gradiente



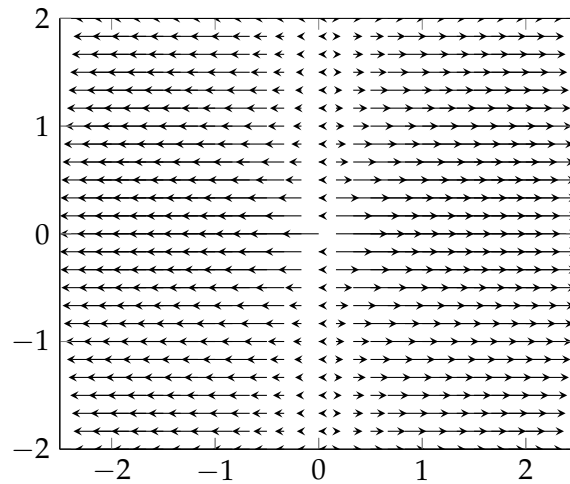
Forma cuadrática para $f(x, y) = x^2$



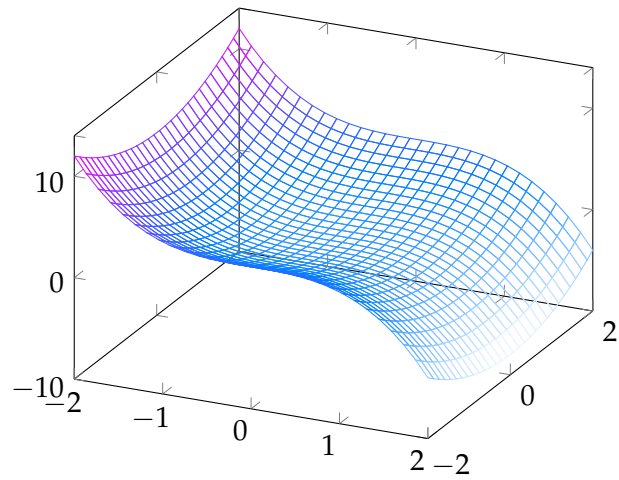
Curvas de nivel para $f(x, y) = x^2$



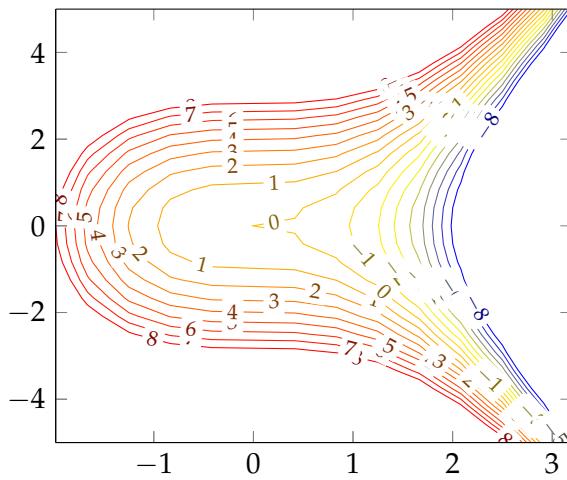
Vector gradiente



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



Curvas de nivel para $f(x,y) = -x^3 + y^2$



Vector gradiente

