1 Derivación en varias variables

[1]:
```
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt
from matplotlib import cm

# Gráficos separados en novas ventás
%matplotlib qt

# Gráficos incrustados neste documento
%matplotlib inline
```

1.1 Gradiente dunha función

[2]:
```
x, y, z = sp.symbols('x y z', real=True)
# Definimos a función
f_xyz = sp.Matrix([x*y**2*z**3])
# Calculamos o seu gradiente
gradiente = f_xyz.jacobian([x,y,z])

print("A función é:")
display(f_xyz)
print("O seu gradiente é")
display(gradiente)
print("Avaliado en (x,y,z)=(1,1,1) é:")
display(gradiente.subs({x:1,y:1,z:1}))

# Calculamos a derivada direccional
dfv = gradiente.subs({x:1,y:1,z:1})@np.array([-1,0,2]).transpose() # O comando @␣
↪realiza o produto de matrices
print('A derivada direccional de f en (x,y,z)=(1,1,1) respecto a v=(-1,0,2) é:␣
↪',dfv[0])
```

```
A función é:
```

$$\begin{bmatrix} xy^2z^3 \end{bmatrix}$$

```
O seu gradiente é
```

$$\begin{bmatrix} y^2z^3 & 2xyz^3 & 3xy^2z^2 \end{bmatrix}$$

```
Avaliado en (x,y,z)=(1,1,1) é:
```

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

```
A derivada direccional de f en (x,y,z)=(1,1,1) respecto a v=(-1,0,2) é:  5
```

1.2 Jacobiana dunha función

[3]:
```
# Definimos a función
f2_xyz = sp.Matrix([x*y*z, sp.sin(x*y ) + z])
# Calculamos a súa jacobiana
jacobiana = f2_xyz.jacobian([x,y,z])

print("A función é:")
display(f2_xyz)
print("O súa jacobiana é")
display(jacobiana)
print("Avaliada en (x,y,z)=(0,1,2) é:")
display(jacobiana.subs({x:0,y:1,z:2}))
```

```
A función é:
```

$$\begin{bmatrix} xyz \\ z + \sin(xy) \end{bmatrix}$$

```
O súa jacobiana é
```

$$\begin{bmatrix} yz & xz & xy \\ y\cos(xy) & x\cos(xy) & 1 \end{bmatrix}$$

```
Avaliada en (x,y,z)=(0,1,2) é:
```

$$\begin{bmatrix} 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

1.3 Hessiana dunha función

[4]:
```
# Definimos a función
f_xy = sp.Lambda((x,y), x*(y+sp.exp(y)))
# Calculamos a súa hessiana
hessiana = sp.hessian(f_xy(x,y), (x,y))

print("A función é:")
display(f_xy)
print("O súa hessiana é")
display(hessiana)
print("Avaliada en (x,y)=(0,1) é:")
display(hessiana.subs({x:0,y:1}))
```

```
A función é:
```

$$((x,\ y) \mapsto x\,(y + e^y))$$

```
O súa hessiana é
```

$$\begin{bmatrix} 0 & e^y + 1 \\ e^y + 1 & xe^y \end{bmatrix}$$

```
Avaliada en (x,y)=(0,1) é:
```

$$\begin{bmatrix} 0 & 1+e \\ 1+e & 0 \end{bmatrix}$$

```
[ ]:
```