Lab 4

Name: (your name)

Some remarks before you start:

- Import the Python packages below each time you start a session.
- Before submitting your report, remove all ouputs by clicking on **Edit→Clear all ouputs**. Then download your report by clicking on **File→Download→.ipynb**. This is the file you will submit on Canvas.

In this lab, you will practice:

- graph functions of two variables and their level sets
- use Gradient Descent/Ascent method to find local min/max

Problems	Points
1,2,3	5
4, 5	6
Readability of your report	3
Total: 5	Total: 30

I. Import necessary Python packages

Execute the following code each time you start a session.

from matplotlib.pyplot import*
from matplotlib.pyplot import contour as contourplot
from matplotlib.pyplot import scatter as scatterplot
from plotly.graph_objects import*
from autograd.numpy import*
from autograd import elementwise_grad as diff

II. Graph a multivariable function and its level sets

To define a function on Python, you will use the keyword **def**. For example, the code below defines the function $f(x,y) = \cos(x)\sin(y)$

```
def f(x,y):
    return cos(x)*sin(y)
```

To graph the function, you first need to generate an array of x-values and an array of y-values using the command **linspace**. For example,

```
x = linspace(-5, 5, 26) # generate 26 evenly-spaced points between -5 and 5
y = linspace(-5, 5, 26)
```

Next, you "mesh" the array of x-values and the array of y-values on the coordinate system. Then you create a matrix of z-values.

```
X, Y = meshgrid(x, y)
Z = f(X,Y)
```

There are two packages of Python that are commonly used for graphing a multivariable function, namely **plotly** and **matplotlib**. Plotly has interactive features that allow you to drag, rotate, scale a graph. Matplotlib's graphs are static but are more customizable. With **plotly**, you graph the function f as follows. You can interact with the graph with your mouse.

fig = Figure(data=[Surface(z=Z, x=X, y=Y)])
fig.update_layout(title='\$f(x, y) = \cos x\sin y\$')

With **matplotlib**, you graph the function f as follows.

```
fig = figure().add_subplot(projection='3d')
fig.plot_surface(X, Y, Z)
# Set viewpoint (optional), where elev is elevation and azim is azimuth angle (in degree):
fig.view_init(elev=55, azim=40)
```

Suppose you want to sketch a contour map of *c*-level curves, where *c* ranges from -2 to 2 with a step size of 0.2. With **plotly**, you will run the following. *Note that you can use your mouse to zoom in any region (drag with left click) or zoom out (double click).*

```
fig2 = Figure(data=Contour(z=Z,x=x,y=y,contours=dict(start=-2.0,end=2.0,size=0.2)))
fig2.update_layout(width=600,height=600)
```

With matplotlib, you run the following:

```
contourplot(X, Y, Z, levels=linspace(-2,2,21), cmap='viridis')
colorbar()
```

Exercise 1

A tumor in a patient's brain is detected by an MRI scan, which models density as a function of position:

$$f(x,y)=e^{-2(x^2+y^2)}+0.3\,e^{-(x^2+y^2)}$$

where D(x, y) represents tissue density at point (x, y).

- Graph the density function to visualize the tumor shape.
- Find level curves for different densities and determine the tumor boundary.
- If the tumor boundary is determined to be where the tissue density is equal to 0.15, find the area of this tumor.

Exercise 2

Meteorologists model temperature (in Celsius) as a function of latitude and longitude:

$$T(x,y) = 20 - 0.01x^2 - 0.02y^2 + 5\sin(0.1x)\cos(0.1y)$$

where x and y represent position coordinates (kilometers from a reference point).

- Graph this temperature distribution in 3D.
- Sketch level curves (isotherms) for specific temperatures $T=10^{\circ}C, 15^{\circ}C, 20^{\circ}C.$
- Interpret how temperature changes in different regions.

III. Gradient Descent/Ascent algorithm for optimization

The *Gradient Descent method* is a simple algorithm to find a local minimum of a function f. You start with an initial guess (x_0, y_0) and successively update it using the recursion

$$(x_{n+1},y_{n+1})=(x_n,y_n)-lpha
abla f(x_n,y_n)$$

This vector-equation is equivalent to a system of two equations:

$$egin{aligned} x_{n+1} &= x_n - lpha f_x(x_n,y_n) \ y_{n+1} &= y_n - lpha f_y(x_n,y_n) \end{aligned}$$

The *Gradient Ascent method* is to find a local maximum. It uses the same recursion formulas as above, except that all minus signs are changed to plus signs. In practice, the common stopping condition is when the magnitude of the gradient vector $\nabla f(x_n, y_n)$ is under a certain threshold

$$|
abla f(x_n,y_n)|=\sqrt{f_x(x_n,y_n)^2+f_y(x_n,y_n)^2}<\epsilon$$

or when (x_n, y_n) falls out of a certain range of interest.

The Python code to compute (x_0, y_0) , (x_1, y_1) , (x_2, y_2) ,..., (x_n, y_n) in the Gradient Descent method is as follows.

```
def gradd(f,x0,y0,alpha,n):
    dfdx = diff(f, argnum=0)
    dfdy = diff(f, argnum=1)
    x = float(x0)
    y = float(y0)
    print(0, x, y, f(x,y), sqrt(dfdx(x,y)**2+dfdy(x,y)**2))
    points=[(x,y)]
    for i in range(n):
        xold = x
        x = x - alpha*dfdx(x,y)
```

```
y = y - alpha*dfdy(xold,y)
points.append((x,y))
print(i+1, x, y, f(x,y), sqrt(dfdx(x,y)**2+dfdy(x,y)**2))
return points
```

For illustration purposes, let us consider

$$f(x,y) = \cos(x)\sin(y)$$

```
def f(x,y):
    return cos(x)*sin(y)
x = linspace(-5, 5, 26)
y = linspace(-5, 5, 26)
X, Y = meshgrid(x, y)
Z = f(X,Y)
```

With the initial guess $(x_0, y_0) = (1, 2)$ and learning rate $\alpha = 0.1$, run the following to get a list of points (x_0, y_0) , (x_1, y_1) , (x_2, y_2) ,..., (x_5, y_5) in the Gradient Descent method.

```
points=gradd(f,1,2,0.1,5)
```

With plotly, you can sketch the points on top of the contour map as follows:

With matplotlib, you sketch the points on top of the contour map using the following code:

```
x, y = zip(*points)
contourplot(X, Y, Z, levels=linspace(-2,2,11), cmap='viridis')
colorbar()
scatterplot(x=x,y=y,s=10) # s is the size of each dot
for i, (xn, yn) in enumerate(points):
    text(xn, yn, str(i), fontsize=9, ha='right', va='bottom', color="red")
```

Exercise 3

- Explain what each line of the function gradd defined above.
- Explain what goes wrong if you replace the three lines:

xold = x x = x - alpha*dfdx(x,y) y = y - alpha*dfdy(xold,y)

by the two lines:

x = x - alpha*dfdx(x,y)
y = y - alpha*dfdy(x,y)

Exercise 4

Let $f(x,y)=x^3+xy^2-6x+y^2$

- Graph the function on the square $(x,y)\in [-3,3] imes [-3,3].$
- Sketch the contour map.
- Use the Gradient Descent/Ascent method to find the maximum and minimum values of f in this square and find the values of (x, y) where they are attained. Stop when either (x_n, y_n) falls out of the square or $|\nabla f(x_n, y_n)| < 0.01$.
- Visualize your sequence of (x_n, y_n) as illustrated above.

Exercise 5

With the same function as in Exercise 4,

- Use the Gradient Descent/Ascent method to find the maximum and minimum values of f in the circle $x^2 + y^2 \le 9$ and find the values of (x, y) where they are attained. Stop when either (x_n, y_n) falls out of the circle or $|\nabla f(x_n, y_n)| < 0.01$.
- Visualize your sequence of (x_n, y_n) as illustrated above.
- Illustrate an example of initial guess and learning rate such that the sequence eventually escapes the circle. Visualize this sequence.