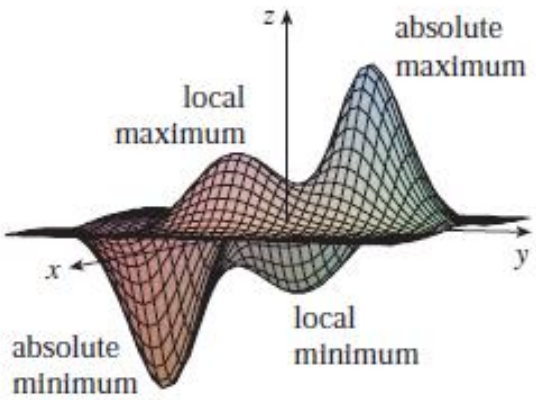
Local Maxima & Local Minima



SECOND DERIVATIVES TEST Suppose the second partial derivatives of f are continuous on a disk with center (a, b), and suppose that $f_x(a, b) = 0$ and $f_y(a, b) = 0$ [that is, (a, b) is a critical point of f]. Let

$$D = D(a, b) = f_{xx}(a, b)\, f_{yy}(a, b) - [f_{xy}(a, b)]^2$$

(a) If $D > 0$ and $f_{xx}(a, b) > 0$, then $f(a, b)$ is a local minimum.

(b) If $D > 0$ and $f_{xx}(a, b) < 0$, then $f(a, b)$ is a local maximum.

(c) If $D < 0$, then $f(a, b)$ is not a local maximum or minimum.

NOTE 1 In case (c) the point (a, b) is called a **saddle point** of f and the graph of f crosses its tangent plane at (a, b).

NOTE 2 If $D = 0$, the test gives no information: f could have a local maximum or local minimum at (a, b), or (a, b) could be a saddle point of f.

NOTE 3 To remember the formula for D, it's helpful to write it as a determinant:

$$D = \begin{vmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{vmatrix} = f_{xx} f_{yy} - (f_{xy})^2$$

Aim

- To write MATLAB codes to evaluate and visualize the critical points, maximum point, minimum points and saddle points of the function $f(x, y)$.

MATLAB Syntax Used:

diff (f)	Differentiate the function with respect to x symbolically
solve(eq)	The input to solve can be either symbolic expressions or strings. If eq is a symbolic expression (x^2 - 2*x + 1) or a string that does not contain an equal sign ('x^2 - 2*x + 1'), then solve(eq) solves the equation $eq = 0$ for its default variable (as determined by symvar).
R = subs(S, old, new)	Replaces old with new in the symbolic expression S.
sprintf (format, A, ...)	Applies the *format* to all elements of array *A* and any additional array arguments in column order, and returns the results to string *str*.
ezsurf (fun)	Creates a graph of fun(x, y) using the surf function. Function is plotted over the default domain: $-2\pi < x < 2\pi$, $-2\pi < y < 2\pi$.
plot3(X1, Y1, Z1,...)	Displays a three-dimensional plot of a set of data points.
shading interp	Varies the color in each line segment and face by interpolating the colormap index or true color value across the line or face.
summer	Consists of colors that are shades of green and yellow.

```
clc
clear all
syms x y real
f= input('Enter the function f(x,y):');
fx= diff(f,x);
fy=diff(f,y);
[ax ay] = solve(fx,fy);
fxx= diff(fx,x);
fxy=diff(fx,y);
fyy =diff(fy,y);
D=fxx*fyy - fxy^2;
```

```
for i = 1:1:size(ax)
    figure
    T1 = subs(subs(D, x ,ax(i) ), y, ay(i));
    T2= subs(subs(fxx, x, ax(i)), y, ay(i));
    T3= subs(subs(f, x, ax(i)), y, ay(i));
```

```
if(double(T1)==0)
    sprintf('The point (x,y) is (%d, %d) and need further
investigation',double (ax(i)), double(ay(i)))
      elseif(double(T1 )< 0)
    sprintf('The point (x,y) is (%d, %d) is saddle point', double(ax(i)),
double(ay(i)))
      else
if(double(T2) < 0)
    sprintf('The maximum point (x,y) is (%d, %d)', double(ax(i)),
double(ay(i)))
    sprintf('The value of the function is %d', double (T3))
      else
    sprintf('The minimum point (x,y) is (%d, %d)', double(ax(i)),
double(ay(i)))
    sprintf('The value of the function is %d', double (T3))
end
end
```

```
ezsurf(f, [double(ax(i))-2, double(ax(i))+2, double(ay(i))-2,
double(ay(i))+2]);
hold on
plot3(double(ax(i)) , double(ay(i)) , double(T3), 'r*', 'markersize', 15);
end
```

Example 1:

Investigate maximum and minimum for the following function $F(x,y)=x^4+y^4-4xy+1$

Example 2:

Find the maximum and minimum values of the function $F(x,y)= x^3y+12x^2-8y$.

Practice Problems:

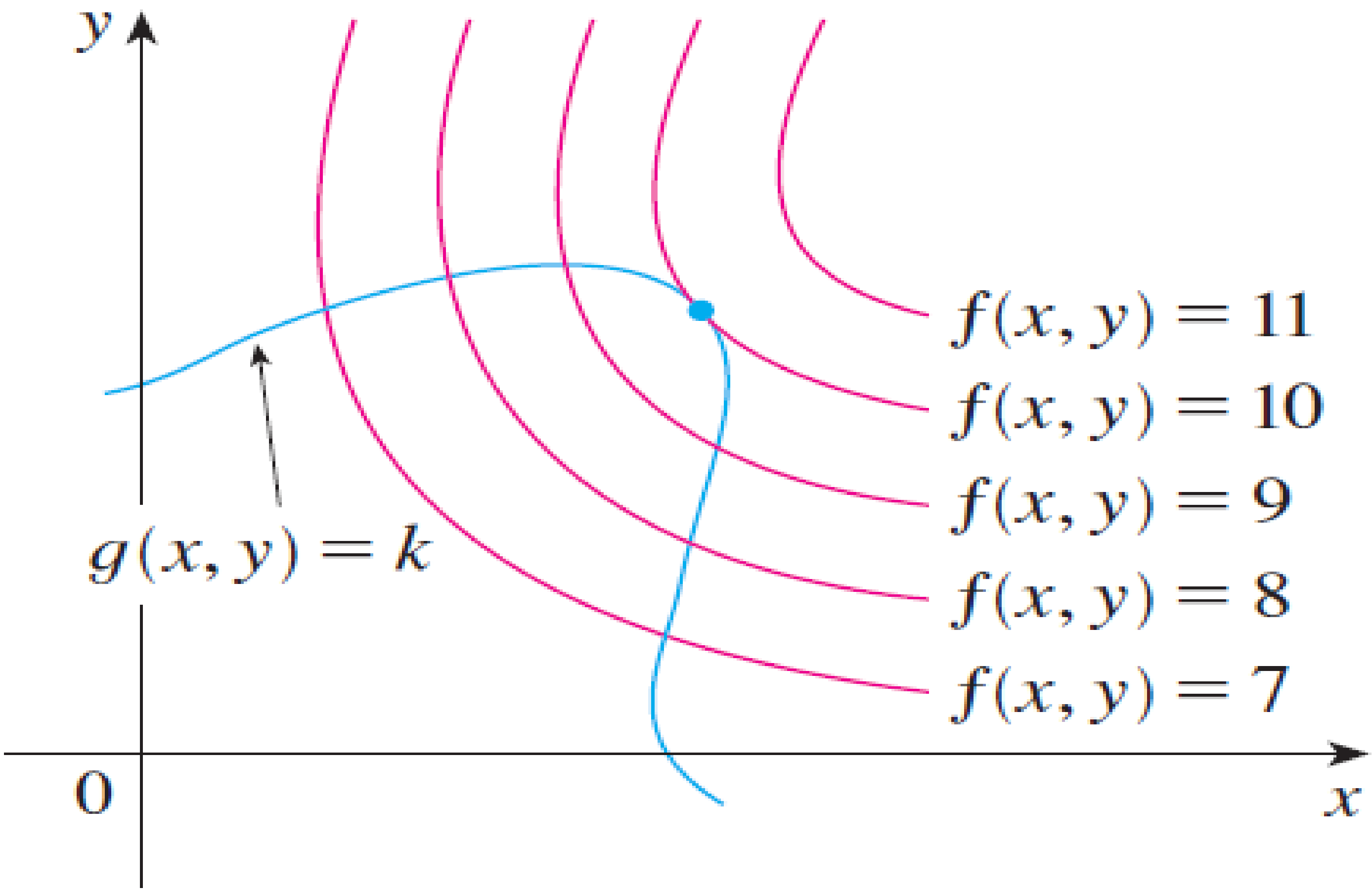
1) Find the maximum and minimum value of the following function

$F(x,y)= 2x^3+xy^2+5x^2+y^2$

2) Investigate the maximum and minimum of the following function

$F(x,y)=x^3y+12x^2$

LAGRANGE MULTIPLIERS



It's easier to explain the geometric basis of Lagrange's method for functions of two variables. So we start by trying to find the extreme values of $f(x, y)$ subject to a constraint of the form $g(x, y) = k$. In other words, we seek the extreme values of $f(x, y)$ when the point (x, y) is restricted to lie on the level curve $g(x, y) = k$. Figure 1 shows this curve together with several level curves of f. These have the equations $f(x, y) = c$, where $c = 7$, 8, 9, 10, 11. To maximize $f(x, y)$ subject to $g(x, y) = k$ is to find the largest value of c such that the level curve $f(x, y) = c$ intersects $g(x, y) = k$. It appears from Figure 1 that this happens when these curves just touch each other, that is, when they have a common tangent line. (Otherwise, the value of c could be increased further.) This means that the normal lines at the point (x_0, y_0) where they touch are identical. So the gradient vectors are parallel; that is, $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$ for some scalar λ.

METHOD OF LAGRANGE MULTIPLIERS To find the maximum and minimum values of $f(x, y, z)$ subject to the constraint $g(x, y, z) = k$ [assuming that these extreme values exist and $\nabla g \neq \mathbf{0}$ on the surface $g(x, y, z) = k$]:

(a) Find all values of x, y, z, and λ such that

$$\nabla f(x, y, z) = \lambda \nabla g(x, y, z)$$

and $$g(x, y, z) = k$$

(b) Evaluate f at all the points (x, y, z) that result from step (a). The largest of these values is the maximum value of f; the smallest is the minimum value of f.

If we write the vector equation $\nabla f = \lambda \nabla g$ in terms of its components, then the equations in step (a) become

$$f_x = \lambda g_x \qquad f_y = \lambda g_y \qquad f_z = \lambda g_z \qquad g(x, y, z) = k$$

This is a system of four equations in the four unknowns x, y, z, and λ, but it is not necessary to find explicit values for λ.

DEFINITION The **gradient vector (gradient)** of $f(x, y)$ at a point $P_0(x_0, y_0)$ is the vector

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j}$$

obtained by evaluating the partial derivatives of f at P_0.

LAGRANGE'S MULTIPLIERS METHOD

Aim:

- To write the MATLAB codes to find maximum and minimum in Lagrange's multipliers method.

MATLAB Syntax used:

diff(f)	Differentiate the function with respect to x symbolically
solve(eq)	The input to solve can be either symbolic expressions or strings. If eq is a symbolic expression (x^2 - 2*x + 1) or a string that does not contain an equal sign ('x^2 - 2*x + 1'), then solve(eq) solves the equation eq = 0 for its default variable (as determined by symvar).
R = subs(S, old, new)	Replaces old with new in the symbolic expression S.
sprintf(*format*, *A*, ...)	Applies the *format* to all elements of array *A* and any additional array arguments in column order, and returns the results to string *str*.
ezsurf(fun)	Creates a graph of fun(x,y) using the surf function. Function is plotted over the default domain: $-2\pi < x < 2\pi$, $-2\pi < y < 2\pi$.
plot3(X1,Y1,Z1,...)	Displays a three-dimensional plot of a set of data points.

```
clc
clear all
syms x y lam real
f= input('Enter the function in terms of x and y:');
g= input('Enter the constrant function in terms of x
and y:');
[alam,ax,ay]=solve(jacobian(f-lam*g,[x y lam]))
T = subs(f,{x,y},{ax,ay})
for i = 1:1:size(T)
figure
sprintf('The point(x,y) is
(%d,%d)',double(ax(i)),double(ay(i)))
sprintf('The value of the function is %d',double(T(i)))
```

```
[X1,Y1]= meshgrid(double(ax(i))-
3:.2:double(ax(i))+3,double(ay(i))-
3:.2:double(ay(i))+3);
zfun = @(x, y) eval(vectorize(f));
Z1=zfun(X1,Y1);
contour(X1,Y1,Z1,50)
hold on
h = ezplot(g,[double(ax(i))-3,double(ax(i))+3]);
set(h,'Color',[1,0.7,0.9])
plot(double(ax(i)),double(ay(i)),'r.','markersize',12)
end
```

Example 1:

Find the extreme values of the function $f(x,y)=x^2-y^2$ subject to the constraints $2y-x^2=0$

Find the extreme values of the function $f(x,y)=2x+2xy+y$ subject to the constraints $2x+y=100$