



Convex Optimization for Machine Learning

with Mathematica Applications

Chapter 3

Multi-Variable Optimization Without Constraints

M. M. Hammad

Department of Mathematics
Faculty of Science
Damanhour University
Egypt

Chapter 3

Multi-Variable Optimization Without Constraints

3.1 Level Curves and Local Linearization of the Two Variables Functions

One way to visualize a function of two variables is through its graph. The graph of f is the surface with equation $z = f(x, y)$. See for example [figure 3.1](#).

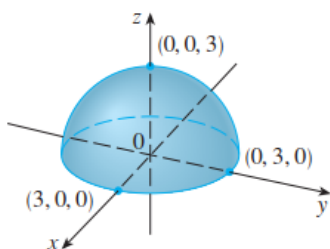


Figure 3.1. The graph of $f(x, y) = \sqrt{9 - x^2 - y^2}$

Another method for visualizing functions is a contour map on which points of constant elevation are joined to form contour lines, or level curves.

Definition (level curves): The level curves of a function of two variables are the curves in the xy -plane with equations $f(x, y) = k$, where k is a constant in the range of f .

More generally, a contour line for a function of two variables is a curve connecting points where the function has the same particular value (a constant value). It is a plane section of the three-dimensional graph of the function $f(x, y)$ parallel to the (x, y) -plane. The surface is steep where the level curves are close together. It is somewhat flatter where they are farther apart, [figure 3.2](#).

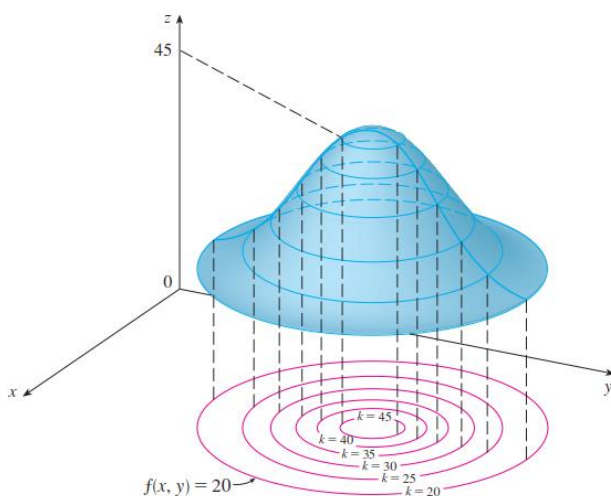


Figure 3.2. The level curve C with equation $f(x, y) = k$ is the projection of the trace of f in the plane $z = k$ onto the xy -plane.

Example 3.1

Sketch a contour map for the surface described by $f(x, y) = x^2 + y^2$, using the level curves corresponding to $k = 0, 1, 4, 9$ and 16 .

Solution

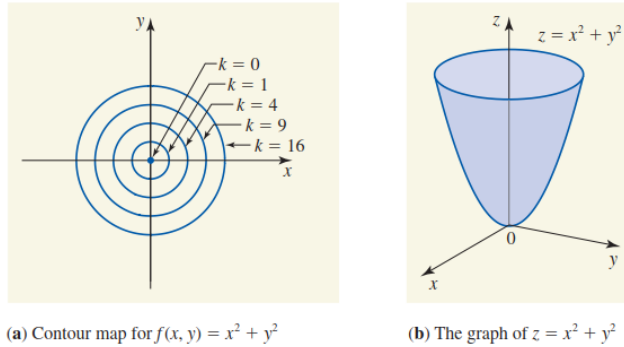


Figure 3.3.

The level curve of f corresponding to each value of k is a circle $x^2 + y^2 = k$ of radius \sqrt{k} , centered at the origin, [figure 3.3](#). For example, if $k = 4$, the level curve is the circle with equation $x^2 + y^2 = 4$, centered at the origin and having radius 2.

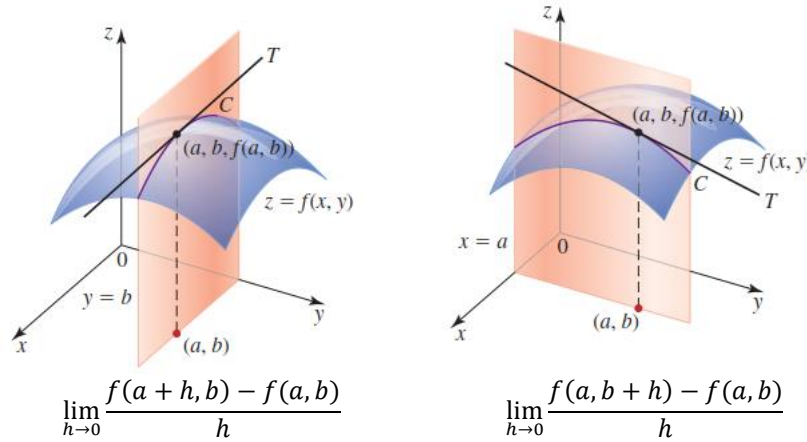
Definition (partial derivative): Let $z = f(x, y)$. Then the partial derivative of f with respect to x is

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x + h, y) - f(x, y)}{h} \quad (3.1)$$

and the partial derivative of f with respect to y is

$$\frac{\partial f}{\partial y} = \lim_{h \rightarrow 0} \frac{f(x, y + h) - f(x, y)}{h} \quad (3.2)$$

provided that each limit exists. (See [figure 3.4](#))



measures the slope of T and the rate of change of $f(x, y)$ in the x -direction when $x = a$ and $y = b$.

measures the slope of T and the rate of change of $f(x, y)$ in the y -direction when $x = a$ and $y = b$.

Figure 3.4.

A plane in space is uniquely determined by specifying a point $P(x_0, y_0, z_0)$ lying in the plane and a vector $\mathbf{n} = (A, B, C)$ that is normal (perpendicular) to it. The general equation of this plane is

$$A(x - x_0) + B(y - y_0) + C(z - z_0) = 0. \quad (3.3)$$

By dividing this equation by C and letting $a = -\frac{A}{C}$ and $b = -\frac{B}{C}$, we can write it in the form

$$z - z_0 = a(x - x_0) + b(y - y_0). \quad (3.4)$$

If (3.4) represents the tangent plane at P , then its intersection with the plane $y = y_0$ must be the tangent line T_1 . Setting $y = y_0$ in (3.4) gives

$$z - z_0 = a(x - x_0), \quad y = y_0, \quad (3.5)$$

and we recognize these as the equations (in point-slope form) of a line with slope a . But we know that the slope of the tangent T_1 is $f_x(x_0, y_0)$. Therefore, $a = f_x(x_0, y_0)$. Similarly, putting $x = x_0$ in (3.4), we get $z - z_0 = b(y - y_0)$, which must represent the tangent line T_2 , so $b = f_y(x_0, y_0)$. (See figure 3.5)

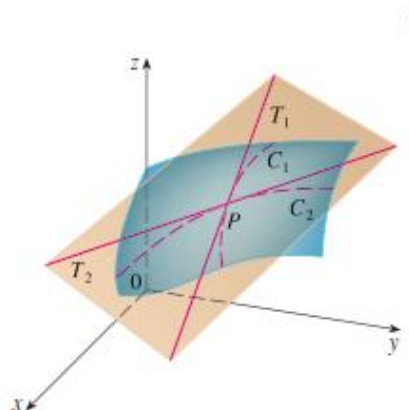


Figure 3.5. The tangent plane contains the tangent lines T_1 and T_2 .

Suppose f has continuous partial derivatives. An equation of the tangent plane to the surface $z = f(x, y)$ at the point $P(x_0, y_0, z_0)$ is

$$z - z_0 = f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0). \quad (3.6)$$

For a function of one variable, local linearity means that as we zoom in on the graph, it looks like a straight line. As we zoom in on the graph of a two-variable function, the graph usually looks like a plane, which is the graph of a linear function of two variables. (See figure 3.6)

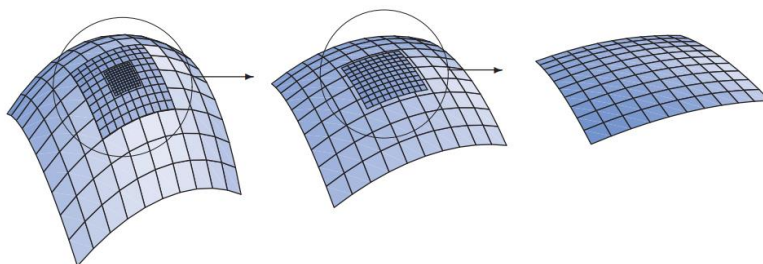


Figure 3.6. Zooming in on the graph of a function of two variables until the graph looks like a plane

Seeing a plane when we zoom in at a point tells us that $f(x, y)$ is closely approximated near that point by a linear function, $L(x, y)$:

$$f(x, y) \approx L(x, y). \quad (3.7)$$

The graph of the function $z = L(x, y)$ is the tangent plane at that point. (See figure 3.7)

Definition (tangent plane): Assuming f is differentiable at (a, b) , the equation of the tangent plane is

$$z = f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b). \quad (3.8)$$

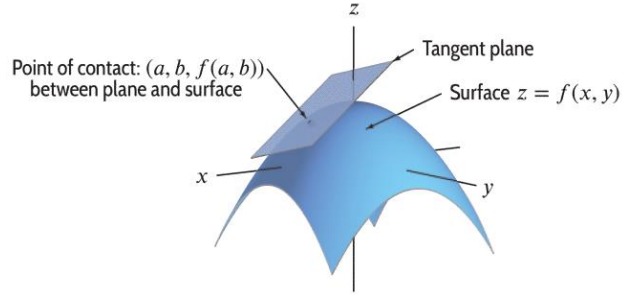


Figure 3.7. The tangent plane to the surface $z = f(x, y)$ at the point (a, b)

Since the tangent plane lies close to the surface near the point at which they meet, z -values on the tangent plane are close to values of $f(x, y)$ for points near (a, b) . Thus, replacing z by $f(x, y)$ in the equation of the tangent plane, we get the following approximation:

Definition (tangent plane): Provided f is differentiable at (a, b) , the approximation $f(x, y)$:

$$f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b). \quad (3.9)$$

is called the linear approximation or the tangent plane approximation of $f(x, y)$.

We are thinking of a and b as fixed, so the expression on the right side is linear in x and y . The right side of this approximation gives the local linearization of f near $x = a, y = b$. Figure 3.8 shows the tangent plane approximation graphically.

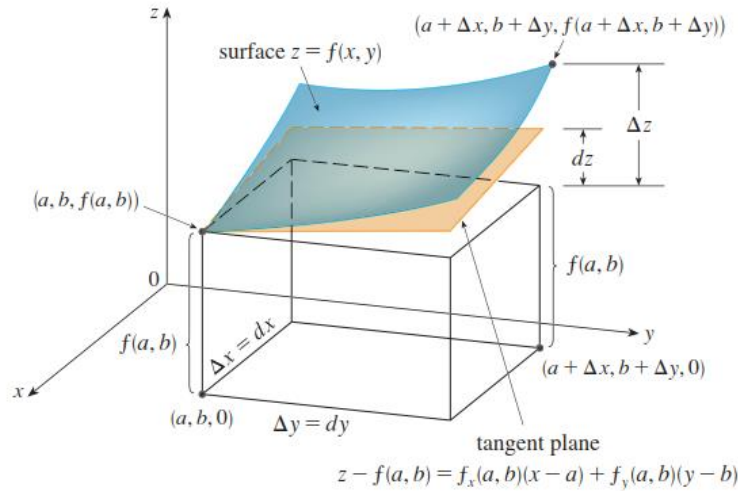


Figure 8. Local linearization: Approximating $f(x, y)$ by the z -value from the tangent plane

Taylor series for a function and linearization

It will be helpful to review the Taylor series for a function of one variable, and see how it extends to functions of more than one variable. Recall that the Taylor series for a function $f(x)$, based at a point $x = a$, is given by the following, where we assume that f is analytic:

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots \quad (3.10)$$

Therefore, we can approximate f using a constant:

$$f(x) \approx f(a) \quad (3.11)$$

or using a linear approximation (which is the tangent line to f at a):

$$f(x) \approx f(a) + f'(a)(x - a) \quad (3.12)$$

or using a quadratic approximation:

$$f(x) \approx f(a) + f'(a)(x - a) + \frac{f''}{2!}(a)(x - a)^2 \quad (3.13)$$

We can do something similar if f depends on more than one variable. For example, if $z = f(x, y)$. In this case, the linearization of f at $x = a, y = b$ is given by the tangent plane:

$$f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) \quad (3.14)$$

If we want to go further with a second order (quadratic) approximation, it looks very similar. First, if $z = f(x, y)$ at (a, b) , the quadratic approximation looks like this:

$$f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) + \frac{1}{2}(f_{xx}(a, b)(x - a)^2 + 2f_{xy}(a, b)(x - a)(y - b) + f_{yy}(a, b)(y - b)^2). \quad (3.15)$$

where we assume that $f_{xy}(a, b) = f_{yx}(a, b)$.

The gradient of f is usually defined as a vector of first partial derivatives:

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (3.16)$$

and the 2×2 matrix of second partial derivatives is called the Hessian matrix

$$\mathbf{H}_f = \begin{pmatrix} f_{xx} & f_{yx} \\ f_{xy} & f_{yy} \end{pmatrix} \quad (3.17)$$

Using this notation, the linear approximation to f at $\hat{\mathbf{x}} = (a, b)^T$ is

$$f(\mathbf{x}) \approx f(\hat{\mathbf{x}}) + \langle \nabla f(\hat{\mathbf{x}}), (\mathbf{x} - \hat{\mathbf{x}}) \rangle \quad (3.18)$$

The quadratic approximation to f is:

$$f(\mathbf{x}) \approx f(\hat{\mathbf{x}}) + \langle \nabla f(\hat{\mathbf{x}}), (\mathbf{x} - \hat{\mathbf{x}}) \rangle + \frac{1}{2} \langle (\mathbf{x} - \hat{\mathbf{x}}) | \mathbf{H}_f(\hat{\mathbf{x}}) | (\mathbf{x} - \hat{\mathbf{x}}) \rangle \quad (3.19)$$

If $f(x, y)$ is a two-dimensional function that has a local extremum at a point (x_0, y_0) and has continuous partial derivatives at this point, then $f_x(x_0, y_0) = 0$ and $f_y(x_0, y_0) = 0$. The second partial derivatives test classifies the point as a local maximum or local minimum.

Definition (second derivative test): let

$$D = f_{xx}f_{yy} - f_{xy}f_{yx} = f_{xx}f_{yy} - f_{xy}^2. \quad (3.20)$$

Suppose that (x_0, y_0) is a critical point of f (that is, $f_x(x_0, y_0) = f_y(x_0, y_0) = 0$). Then the second partial derivative test asserts the following:

1. If $D > 0$ and $f_{xx}(x_0, y_0) > 0$, the point is a local minimum (positive definite).
2. If $D > 0$ and $f_{xx}(x_0, y_0) < 0$, the point is a local maximum (negative definite).
3. If $D < 0$, the point is a saddle point.
4. If $D = 0$, higher order tests must be used.

Definition (saddle point): A saddle point or minimax point is a point on the surface of the graph of a function where the slopes (derivatives) in orthogonal directions are all zero (a critical point), but which is not a local extremum of the function.

In the most general terms, a saddle point for a smooth function (whose graph is a curve, surface or hypersurface) is a stationary point such that the curve/surface/etc. in the neighborhood of that point is not entirely on any side of the tangent space at that point.

- Note that in cases 1 and 2, the requirement that D is positive at (x, y) implies that f_{xx} and f_{yy} have the same sign. Therefore, the second condition, that f_{xx} be greater (or less) than zero, could equivalently be that f_{yy} be greater (or less) than zero at that point.
- $D(x_0, y_0)$ and $f_{xx}(x_0, y_0)$ are the principal minors of the Hessian. The first two conditions listed above on the signs of these minors are the conditions for the positive or negative definiteness of the Hessian.

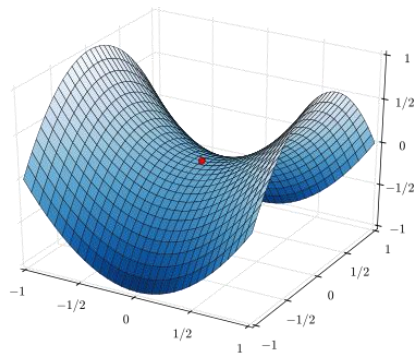


Figure 3.9. A saddle point (in red) on the graph of $z = x^2 - y^2$.

3.2 The Directional Derivative

Suppose that f is a function defined by the equation $z = f(x, y)$, and let $P(a, b)$ be a point in the domain D of f . Furthermore, let \mathbf{u} be a unit (position) vector having a specified direction. Then the vertical plane containing the line L passing through $P(a, b)$ and having the same direction as \mathbf{u} will intersect the surface $z = f(x, y)$ along a curve C (see figure 3.10). Intuitively, we see that the rate of change of z at the point $P(a, b)$ with respect to the distance measured along L is given by the slope of the tangent line T to the curve C at the point $\hat{P}(a, b, f(a, b))$.

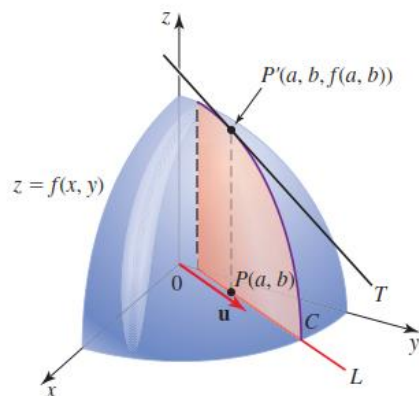


Figure 3.10. The rate of change of z at $P(a, b)$ with respect to the distance measured along L is given by the slope of T .

Let's find the slope of T . First, observe that \mathbf{u} may be specified by writing $\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j}$ for appropriate components u_1 and u_2 . Equivalently, we may specify \mathbf{u} by giving the angle θ that it makes with the positive x -axis, in which case $u_1 = \cos \theta$ and $u_2 = \sin \theta$ (see figure 3.11).

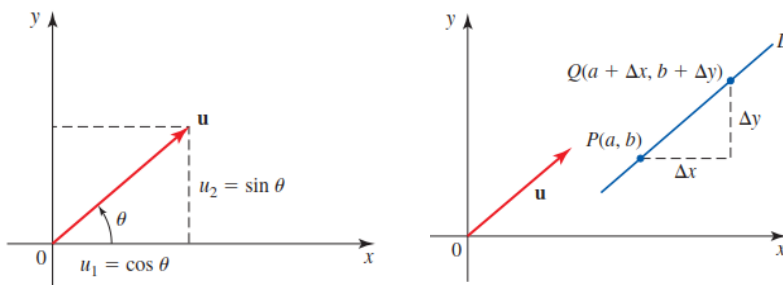


Figure 3.11. Any direction in the plane can be specified in terms of a unit vector \mathbf{u} . The point $Q(a + \Delta x, b + \Delta y)$ lies on L and is distinct from $P(a, b)$.

Next, let $Q(a + \Delta x, b + \Delta y)$ be any point distinct from $P(a, b)$ lying on the line L passing through P and having the same direction as \mathbf{u} (figure 3.12).

Since the vector \mathbf{PQ} is parallel to \mathbf{u} , it must be a scalar multiple of \mathbf{u} . In other words, there exists a nonzero number h such that

$$\mathbf{PQ} = h\mathbf{u} = hu_1 \mathbf{i} + hu_2 \mathbf{j} \quad (3.21)$$

But \mathbf{PQ} is also given by $\Delta x \mathbf{i} + \Delta y \mathbf{j}$, and therefore,

$$\Delta x = hu_1, \quad \Delta y = hu_2, \quad h = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (3.22)$$

So the point Q can be expressed as $Q(a + hu_1, b + hu_2)$. Therefore, the slope of the secant line S passing through the points \hat{P} and \hat{Q} (see figure 3.12) is given by

$$\frac{\Delta z}{h} = \frac{f(a + hu_1, b + hu_2) - f(a, b)}{h} \quad (3.23)$$

Observe that (3.23) also gives the average rate of change of $z = f(x, y)$ from $P(a, b)$ to $Q(a + \Delta x, b + \Delta y) = Q(a + hu_1, b + hu_2)$ in the direction of \mathbf{u} .

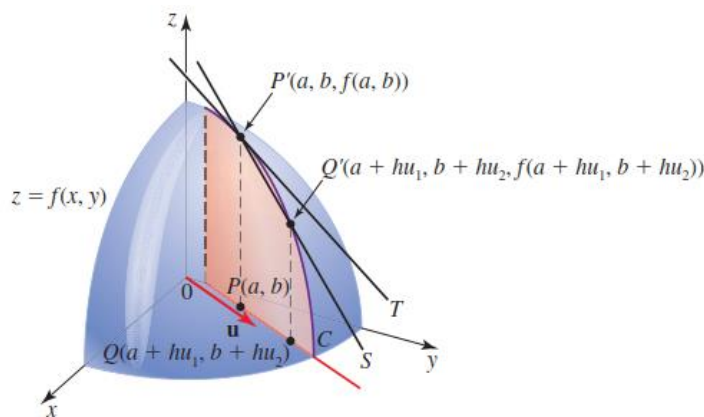


Figure 3.12. The secant line passes through the points \hat{P} and \hat{Q} on the curve C .

If we let h approach zero in (3.23), we see that the slope of the secant line S approaches the slope of the tangent line at \hat{P} . Also, the average rate of change of z approaches the (instantaneous) rate of change of z at (a, b) in the direction of \mathbf{u} . This limit, whenever it exists, is called the directional derivative of f at (a, b) in the direction of \mathbf{u} . Since the point $P(a, b)$ is arbitrary, we can replace it by $P(x, y)$ and define the directional derivative of f at any point as follows.

Definition (directional derivative): Let f be a function of x and y and let $\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j}$ be a unit vector. Then the directional derivative of f at (x, y) in the direction of \mathbf{u} is

$$D_{\mathbf{u}}f(x, y) = \lim_{h \rightarrow 0} \frac{f(x + hu_1, y + hu_2) - f(x, y)}{h} \quad (3.24)$$

if this limit exists.

Note If $\mathbf{u} = \mathbf{i}$ ($u_1 = 1$ and $u_2 = 0$), then (3.24) gives

$$D_{\mathbf{i}}f(x, y) = \lim_{h \rightarrow 0} \frac{f(x + h, y) - f(x, y)}{h} = f_x(x, y) \quad (3.25)$$

That is, the directional derivative of f in the x -direction is the partial derivative of f in the x -direction, as expected. Similarly, you can show that $D_{\mathbf{j}}f(x, y) = f_y(x, y)$.

Theorem 3.1: If f is a differentiable function of x and y , then f has a directional derivative in the direction of any unit vector $\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j}$ and

$$D_{\mathbf{u}}f(x, y) = f_x(x, y)u_1 + f_y(x, y)u_2 \quad (3.26)$$

Proof: Fix the point (a, b) . Using local linearity, we have

$$\Delta f = f(x, y) - f(a, b) \approx f_x(a, b)\Delta x + f_y(a, b)\Delta y = f_x(a, b)hu_1 + f_y(a, b)hu_2 \quad (3.27)$$

Thus, dividing by h gives

$$\frac{\Delta f}{h} \approx f_x(a, b)u_1 + f_y(a, b)u_2 \quad (3.28)$$

This approximation becomes exact as $h \rightarrow 0$, so we have the following formula:

$$D_{\mathbf{u}}f(a, b) = f_x(a, b)u_1 + f_y(a, b)u_2 \quad (3.29)$$

Finally, since (a, b) is arbitrary, we may replace it by (x, y) and the result follows. ■

3.3 The Gradient and Tangent Planes of a Function of Two Variables

The directional derivative $D_{\mathbf{u}}f(x, y)$ can be written as the dot product of the unit vector

$$\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j} \quad (3.30)$$

and the vector

$$f_x(x, y) \mathbf{i} + f_y(x, y) \mathbf{j} \quad (3.31)$$

Thus,

$$D_{\mathbf{u}}f(x, y) = (u_1 \mathbf{i} + u_2 \mathbf{j}) \cdot (f_x(x, y) \mathbf{i} + f_y(x, y) \mathbf{j}) = f_x(x, y)u_1 + f_y(x, y)u_2 \quad (3.32)$$

The vector $f_x(x, y) \mathbf{i} + f_y(x, y) \mathbf{j}$ plays an important role in many other computations and is given a special name.

Definition (gradient): Let f be a function of two variables x and y . The gradient of f is the vector function

$$\nabla f(x, y) = f_x(x, y) \mathbf{i} + f_y(x, y) \mathbf{j} \quad (3.33)$$

Theorem 3.2: If f is a differentiable function of x and y , then f has a directional derivative in the direction of any unit vector \mathbf{u} , and

$$D_{\mathbf{u}}f(x, y) = \langle \nabla f(x, y), \mathbf{u} \rangle \quad (3.34)$$

Suppose θ is the angle between the vectors $\nabla f(x, y)$ and \mathbf{u} . At the point (a, b) , we have

$$\langle \nabla f, \mathbf{u} \rangle = \|\nabla f\| \|\mathbf{u}\| \cos \theta = \|\nabla f\| \cos \theta \quad (3.35)$$

Imagine that ∇f is fixed and that \mathbf{u} can rotate. (see [figure 3.13](#)). The maximum value of $D_{\mathbf{u}}f$ occurs when $\cos \theta = 1$, so $\theta = 0$ and \mathbf{u} is pointing in the direction of $\nabla f(x, y)$.

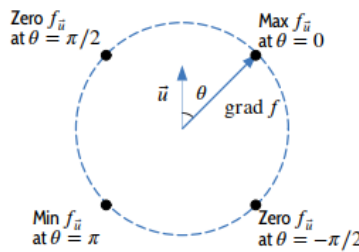


Figure 3.13. Values of the directional derivative at different angles to the gradient

Then

$$\text{Max } D_{\mathbf{u}}f = \|\nabla f\| \quad (3.36)$$

The minimum value of $D_{\mathbf{u}}f$ occurs when $\cos \theta = -1$, so $\theta = \pi$ and \mathbf{u} is pointing in the direction opposite to ∇f . Then

$$\text{Min } D_{\mathbf{u}}f = -\|\nabla f\|. \quad (3.37)$$

Hence, we have

Theorem 3.3: Suppose f is differentiable at the point (x, y) .

1. If $\nabla f(x, y) = \mathbf{0}$, then $D_{\mathbf{u}}f(x, y) = 0$ for every \mathbf{u} .
2. The maximum value of $D_{\mathbf{u}}f(x, y)$ is $|\nabla f(x, y)|$, and this occurs when \mathbf{u} has the same direction as $\nabla f(x, y)$.
3. The minimum value of $D_{\mathbf{u}}f(x, y)$ is $-|\nabla f(x, y)|$, and this occurs when \mathbf{u} has the direction of $-\nabla f(x, y)$.

Notes

1. Property (2) of theorem 3.3 tells us that f increases most rapidly in the direction of $\nabla f(x, y)$. This direction is called the direction of steepest ascent.

2. Property (3) of theorem 3.3 says that f decreases most rapidly in the direction of $-\nabla f(x, y)$. This direction is called the direction of steepest descent.

We are now in a position to give the geometric interpretation of the Gradient. Suppose that the curve C is represented by the vector function

$$\mathbf{r}(t) = g(t)\mathbf{i} + h(t)\mathbf{j} \quad (3.38)$$

where g and h are differentiable functions, $a = g(t_0)$ and $b = h(t_0)$, and t_0 lies in the parameter interval (figure 3.14). Since the point $(x, y) = (g(t), h(t))$ lies on C , we have

$$f(x, y) = f(g(t), h(t)) = c \quad (3.39)$$

for all t in the parameter interval.

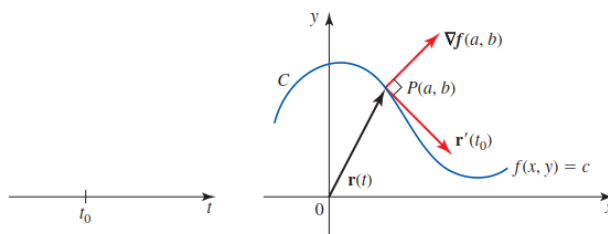


Figure 3.14. The curve may be represented by $\mathbf{r}(t) = x\mathbf{i} + y\mathbf{j} = g(t)\mathbf{i} + h(t)\mathbf{j}$.

Differentiating both sides of this equation with respect to t and using the Chain Rule for a function of two variables, we obtain

$$\frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} = 0 \quad (3.40)$$

Recalling that

$$\nabla f(x, y) = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} \quad (3.41)$$

and

$$\mathbf{r}'(t) = \frac{dx}{dt} \mathbf{i} + \frac{dy}{dt} \mathbf{j} \quad (3.42)$$

we can write this last equation in the form

$$\langle \nabla f(x, y), \mathbf{r}'(t) \rangle = 0 \quad (3.43)$$

In particular, when $t = t_0$, we have

$$\langle \nabla f(a, b), \mathbf{r}'(t_0) \rangle = 0 \quad (3.44)$$

Thus, if $\mathbf{r}'(t_0) \neq \mathbf{0}$, the vector $\nabla f(a, b)$ is orthogonal to the tangent vector $\mathbf{r}'(t_0)$ at $P(a, b)$. Loosely speaking, we have demonstrated the following:

Theorem 3.4.1: ∇f is orthogonal to the level curve $f(x, y) = c$ at P .

Example 3.2

Let $f(x, y) = x^2 - y^2$. Find the level curve of f passing through the point $(5, 3)$. Also, find the gradient of f at that point, and make a sketch of both the level curve and the gradient vector.

Solution

Since $f(5, 3) = 25 - 9 = 16$, the required level curve is the hyperbola $x^2 - y^2 = 16$. The gradient of f at any point (x, y) is

$$\nabla f(x, y) = 2x \mathbf{i} - 2y \mathbf{j}$$

and, in particular, the gradient of f at $(5, 3)$ is

$$\nabla f(5, 3) = 10 \mathbf{i} - 6 \mathbf{j}$$

The level curve and $\nabla f(5, 3)$ are shown in figure 3.15.

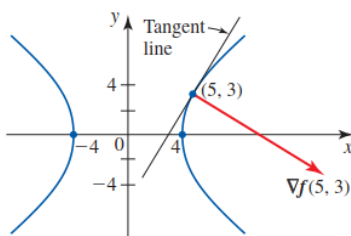


Figure 3.15. The gradient $\nabla f(5, 3)$ is orthogonal to the level curve $x^2 - y^2 = 16$ at $(5, 3)$.

Functions of Three Variables and Level Surfaces

A function f of three variables is a rule that assigns to each ordered triple (x, y, z) in a domain $D = \{(x, y, z) : x, y, z \in \mathbb{R}\}$ a unique real number w denoted by $f(x, y, z) = w$. For example, $f(x, y, z) = xyz$. Since the graph of a function of three variables is composed of the points (x, y, z, w) , where $w = f(x, y, z)$, lying in four-dimensional space, we cannot draw the graphs of such functions. But by examining the level surfaces, which are the surfaces with equations $f(x, y, z) = k$, k a constant, we are often able to gain some insight into the nature of f .

Example 3.3

Find the level surfaces of the function defined by $f(x, y, z) = x^2 + y^2 + z^2$.

Solution

The required level surfaces of f are the graphs of the equations $x^2 + y^2 + z^2 = k$, where $k \geq 0$. These surfaces are concentric spheres of radius \sqrt{k} centered at the origin (see figure 3.16). Observe that k has the same value for all points (x, y, z) lying on any such sphere.

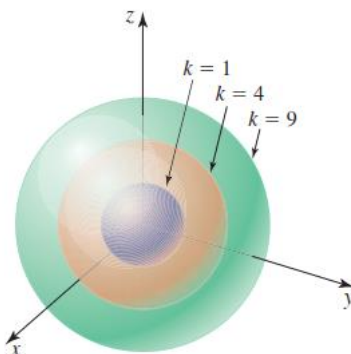


Figure 3.16. The level surfaces of $f(x, y, z) = x^2 + y^2 + z^2$ corresponding to $k = 1, 4, 9$.

Suppose that $F(x, y, z) = k$ is the level surface S of a differentiable function F defined by $T = F(x, y, z)$. Suppose that $P(a, b, c)$ is a point on S and let C be a smooth curve on S passing through P . Then C can be described by the

vector function $\mathbf{r}(t) = f(t)\mathbf{i} + g(t)\mathbf{j} + h(t)\mathbf{k}$ where $f(t_0) = a$, $g(t_0) = b$, $h(t_0) = c$, and t_0 is a point in the parameter interval (see figure 3.17).

Since the point $(x, y, z) = (f(t), g(t), h(t))$ lies on S , we have, $F(x, y, z) = F(f(t), g(t), h(t)) = k$, for all t in the parameter interval. If \mathbf{r} is differentiable, then we can use the Chain Rule to differentiate both sides of this equation to obtain

$$\frac{\partial F}{\partial x} \frac{dx}{dt} + \frac{\partial F}{\partial y} \frac{dy}{dt} + \frac{\partial F}{\partial z} \frac{dz}{dt} = 0 \quad (3.45)$$

This is the same as

$$(F_x(x, y, z)\mathbf{i} + F_y(x, y, z)\mathbf{j} + F_z(x, y, z)\mathbf{k}) \cdot \left(\frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} + \frac{dz}{dt}\mathbf{k} \right) = 0 \quad (3.46)$$

or, in an even more abbreviated form, $\nabla F(x, y, z) \cdot \mathbf{r}'(t) = 0$. In particular, at $t = t_0$ we have $\nabla F(a, b, c) \cdot \mathbf{r}'(t_0) = 0$.

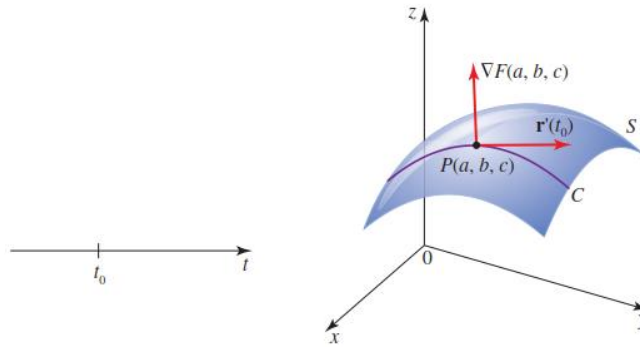


Figure 3.17. The curve C is described by $\mathbf{r}(t) = f(t)\mathbf{i} + g(t)\mathbf{j} + h(t)\mathbf{k}$ with $P(a, b, c)$ corresponding to t_0 .

This shows that if $\mathbf{r}'(t_0) \neq \mathbf{0}$, then the gradient vector $\nabla F(a, b, c)$ is orthogonal to the tangent vector $\mathbf{r}'(t_0)$ to C at P (see figure 3.17). Since this argument holds for any differentiable curve passing through P on C , we have shown that $\nabla F(a, b, c)$ is orthogonal to the tangent vector of every curve on S passing through P . Thus, loosely speaking, we have demonstrated the following result.

Theorem 3.4.2: ∇F is orthogonal to the level surface $F(x, y, z) = 0$ at P .

The gradient $\nabla F(a, b, c)$ at P is orthogonal to the tangent vector of every curve on S passing through P (figure 3.17). This suggests that we define the tangent plane to S at P to be the plane passing through P and containing all these tangent vectors. Equivalently, the tangent plane should have $\nabla F(a, b, c)$ as a normal vector.

Definition (normal line): Let $P(a, b, c)$ be a point on the surface S described by $F(x, y, z) = 0$, where F is differentiable at P , and suppose that $\nabla F(a, b, c) \neq \mathbf{0}$. Then the tangent plane to S at P is the plane that passes through P and has normal vector $\nabla F(a, b, c)$. The normal line to S at P is the line that passes through P and has the same direction as $\nabla F(a, b, c)$.

The equation of the tangent plane is

$$F_x(a, b, c)(x - a) + F_y(a, b, c)(y - b) + F_z(a, b, c)(z - c) = 0 \quad (3.47)$$

3.4 Optimality Criteria

The present section will very largely consist in a generalization of the results of Chapter 2 to the case of more than one variable. We have,

$$\text{optimize: } z = f(\mathbf{x}), \quad \text{where } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (3.48)$$

The definition of a local, a global, or an inflection point remains the same as that for single-variable functions, but the optimality criteria for multivariable functions are different. In a multivariable function, the gradient of a function is not a scalar quantity; instead it is a vector quantity. The optimality criteria can be derived by using the definition of a local optimal point and by using Taylor's series expansion of a multivariable function. Without going into the details of the analysis, we simply present the optimality criteria for a multivariable function. In this chapter and subsequent chapters, we assume that the objective function is a function of n variables represented by x_1, x_2, \dots, x_n .

Definition (ϵ -neighborhood): An ϵ -neighborhood ($\epsilon > 0$) around $\hat{\mathbf{x}}$ is the set of all vectors \mathbf{x} such that

$$\|\mathbf{x} - \hat{\mathbf{x}}\| = \langle (\mathbf{x} - \hat{\mathbf{x}}), (\mathbf{x} - \hat{\mathbf{x}}) \rangle = (x_1 - \hat{x}_1)^2 + (x_2 - \hat{x}_2)^2 + \dots + (x_n - \hat{x}_n)^2 \leq \epsilon^2 \quad (3.49)$$

In geometrical terms, an ϵ -neighborhood around $\hat{\mathbf{x}}$ is the interior and boundary of an n -dimensional sphere of radius ϵ centered at $\hat{\mathbf{x}}$.

An objective function $f(\mathbf{x})$ has a local maximum at $\hat{\mathbf{x}}$ if there exists an ϵ -neighborhood around $\hat{\mathbf{x}}$ such that $f(\mathbf{x}) \leq f(\hat{\mathbf{x}})$ for all \mathbf{x} in this ϵ -neighborhood at which the function is defined. If the condition is met for every positive ϵ (no matter how large), then $f(\mathbf{x})$ has a global maximum at $\hat{\mathbf{x}}$.

Definition (gradient vector): The gradient vector at any point $\hat{\mathbf{x}}$ is represented by $\nabla f(\hat{\mathbf{x}})$ which is an n -dimensional vector given as follows:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \quad (3.50)$$

The notation $\nabla f|_{\hat{\mathbf{x}}}$ signifies the value of the gradient at $\hat{\mathbf{x}}$. The first-order partial derivatives can be calculated numerically using finite difference.

Theorem 3.5: For small displacements from $\hat{\mathbf{x}}$ in various directions, the direction of maximum increase in $f(\hat{\mathbf{x}})$ is the direction of the vector $\nabla f|_{\hat{\mathbf{x}}}$.

Proof:

For any fixed vector $\hat{\mathbf{x}}$ and any unit vector \mathbf{u} , the directional derivative,

$$D_{\mathbf{u}}f(\hat{\mathbf{x}}) = \langle \nabla f|_{\hat{\mathbf{x}}}, \mathbf{u} \rangle$$

gives the rate of change of $f(\hat{\mathbf{x}})$ at $\hat{\mathbf{x}}$ in the direction of \mathbf{u} . Since

$$\langle \nabla f, \mathbf{u} \rangle = \|\nabla f\| \|\mathbf{u}\| \cos \theta = \|\nabla f\| \cos \theta$$

the greatest increase in $f(\mathbf{x})$ occurs when $\theta = 0$, i.e., when \mathbf{u} is in the same direction as ∇f . Therefore, any (small) movement from $\hat{\mathbf{x}}$ in the direction of $\nabla f|_{\hat{\mathbf{x}}}$ will, initially, increase the function over $f(\hat{\mathbf{x}})$ as rapidly as possible. ■

Example 3.4

For $f(x_1, x_2, x_3) = 3x_1^2x_2 - x_2^2x_3^3$ with $\hat{\mathbf{x}} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, find $\nabla f|_{\hat{\mathbf{x}}}$.

Solution

$$\nabla f = \begin{pmatrix} 6x_1x_2 \\ 3x_1^2 - 2x_2x_3^3 \\ -3x_2^2x_3^2 \end{pmatrix}$$

Whence

$$\nabla f|_{\hat{\mathbf{x}}} = \begin{pmatrix} 6(1)(2) \\ 3(1)^2 - 2(2)(3)^3 \\ -3(2)^2(3)^2 \end{pmatrix} = \begin{pmatrix} 12 \\ -105 \\ -108 \end{pmatrix}$$

Therefore, at $([1, 2, 3])^T$, the function increases most rapidly in the direction of $(12, -105, -108)^T$.

Definition (Hessian matrix): The second-order derivatives in multivariable functions form a matrix, $\nabla^2 f(\hat{\mathbf{x}})$ (better known as the Hessian matrix) given as follows:

$$\mathbf{H}_f \equiv \nabla^2 f(\hat{\mathbf{x}}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \\ \vdots & \vdots & \ddots & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right), \quad (i, j = 1, 2, \dots, n) \quad (3.51)$$

The notation $\mathbf{H}_f|_{\hat{\mathbf{x}}}$ signifies the value of the Hessian matrix at $\hat{\mathbf{x}}$. The second-order partial derivatives can also be calculated numerically.

By defining the derivatives, we are now ready to present the optimality criteria:

Definition (negative definite): An $n \times n$ symmetric matrix A (one such that $A = A^T$) is positive definite (positive semi-definite) if $\langle \mathbf{x}|A|\mathbf{x} \rangle$ is positive (non-negative) for every n -dimensional vector $\mathbf{x} \neq \mathbf{0}$.

There are a number of ways to investigate whether a matrix is positive-definite. One common way to do this is to evaluate the eigenvalues of the matrix. If all eigenvalues are positive, the matrix is positive-definite. The other way to test the positive-definiteness of a matrix is to calculate the principal determinants of the matrix. If all principal determinants are positive, the matrix is positive-definite. It is worth mentioning here that the negative-definiteness of a matrix A can be verified by testing the positive-definiteness of the matrix $-A$.

Theorem 3.6: Let $A = [a_{ij}]$ be an $n \times n$ symmetric matrix, and define the determinants

$$A_1 = |a_{11}|, A_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, A_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \dots, A_n \equiv (-1)^{n-1} \det A \quad (3.52)$$

Then A is negative definite if and only if A_1, A_2, \dots, A_n are all negative; A is negative semi-definite if and only if A_1, A_2, \dots, A_r ($r < n$) are all negative and the remaining A 's are all zero.

Example 3.5

For $f(x_1, x_2, x_3) = 3x_1^2x_2 - x_2^2x_3^3$ with $\hat{\mathbf{x}} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, find $\mathbf{H}_f|_{\hat{\mathbf{x}}}$.

Solution

$$\mathbf{H}_f = \begin{pmatrix} 6x_2 & 6x_1 & 0 \\ 6x_1 & -2x_3^3 & -6x_2x_3^2 \\ 0 & -6x_2x_3^2 & -6x_2^2x_3 \end{pmatrix}$$

whence

$$\mathbf{H}_f|_{\hat{\mathbf{x}}} = \begin{pmatrix} 12 & 6 & 0 \\ 6 & -54 & -108 \\ 0 & -108 & -72 \end{pmatrix}$$

For $\mathbf{H}_f|_{\hat{\mathbf{x}}}$, $A_1 = 12 > 0$, so that \mathbf{H}_f is not negative definite, or even negative semi-definite, at $\hat{\mathbf{x}}$.

Theorem 3.7: If $f(\mathbf{x})$ is continuous on a closed and bounded region, then $f(\mathbf{x})$ has a global maximum (and also a global minimum) on that region.

Theorem 3.8: If $f(\mathbf{x})$ has a local maximum (or a local minimum) at \mathbf{x}^* and if ∇f exists on some ϵ -neighborhood around \mathbf{x}^* , then $\nabla f|_{\mathbf{x}^*} = 0$.

Theorem 3.9: If $f(\mathbf{x})$ has second partial derivatives on an ϵ -neighborhood around \mathbf{x}^* , and if $\nabla f|_{\mathbf{x}^*} = 0$ and $\mathbf{H}_f|_{\mathbf{x}^*}$ is positive-definite, then $f(\mathbf{x})$ has a local minimum at \mathbf{x}^* .

It follows from theorems 3.7 and 3.8 that a continuous $f(\mathbf{x})$ assumes its global maximum among those points at which ∇f does not exist or among those points at which $\nabla f = 0$ (stationary points). (A point \mathbf{x} is a stationary point if $\nabla f(\mathbf{x}) = 0$. Furthermore, the point is a minimum, a maximum, or an inflection point if $\nabla^2 f(\mathbf{x})$ is positive-definite, negative-definite, or otherwise, respectively.)

Analytical solutions based on calculus are even harder to obtain for multivariable programs than for single-variable programs, and so, once again, numerical methods are used to approximate (local) maxima to within prescribed tolerances.

3.5 Numerical Algorithms

Recall that a level set of a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the set of points \mathbf{x} satisfying $f(\mathbf{x}) = c$ for some constant c . Thus, a point $\mathbf{x}_0 \in \mathbb{R}^n$ is on the level set corresponding to level c if $f(\mathbf{x}_0) = c$. In the case of functions of two real variables, $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, the notion of the level set is illustrated in Figure 3.18.

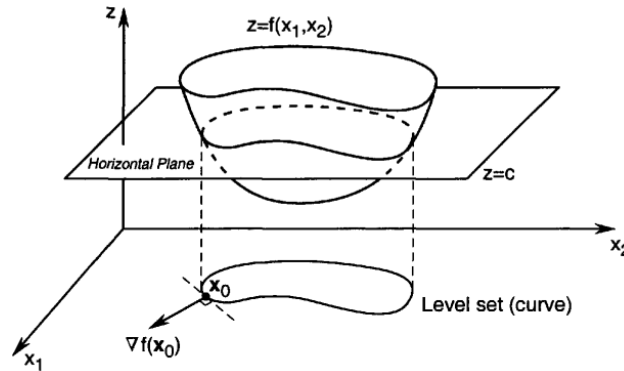


Figure 3.18. Constructing a level set corresponding to level c for f .

The gradient of f at \mathbf{x}_0 , $\nabla f(\mathbf{x}_0)$, if it is not a zero vector, is orthogonal to the tangent vector to an arbitrary smooth curve passing through \mathbf{x}_0 on the level set $f(\mathbf{x}) = c$. Thus, the direction of maximum rate of increase of a real-valued differentiable function at a point is orthogonal to the level set of the function through that point. In other words, the gradient acts in such a direction that for a given small displacement, the function f increases more in the direction of the gradient than in any other direction. To prove this statement, recall that $\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle$, $\|\mathbf{d}\| = 1$, is the rate of increase of f in the direction \mathbf{d} at the point \mathbf{x} . By the Cauchy-Schwarz inequality,

$$\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle \leq \|\nabla f(\mathbf{x})\| \quad (3.53)$$

because $\|\mathbf{d}\| = 1$. But if $\mathbf{d} = \nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$, then

$$\langle \nabla f(\mathbf{x}), \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} \rangle = \frac{1}{\|\nabla f(\mathbf{x})\|} \langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle = \frac{\|\nabla f(\mathbf{x})\|^2}{\|\nabla f(\mathbf{x})\|} = \|\nabla f(\mathbf{x})\| \quad (3.54)$$

Thus, the direction in which $\nabla f(\mathbf{x})$ points is the direction of maximum rate of increase of f at \mathbf{x} . The direction in which $-\nabla f(\mathbf{x})$ points is the direction of maximum rate of decrease of f at \mathbf{x} . Hence, the direction of negative gradient is a good direction to search if we want to find a function minimizer.

We proceed as follows. Let $\mathbf{x}^{(0)}$ be a starting point, and consider the point $\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$. Then, by Taylor's theorem we obtain

$$f(\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})) = f(\mathbf{x}^{(0)}) + \langle \nabla f(\mathbf{x}^{(0)}), -\alpha \nabla f(\mathbf{x}^{(0)}) \rangle + \dots = f(\mathbf{x}^{(0)}) - \alpha \|\nabla f(\mathbf{x}^{(0)})\|^2 + \dots \quad (3.55)$$

Thus, if $\nabla f(\mathbf{x}^{(0)}) \neq 0$, then for sufficiently small $\alpha > 0$, we have

$$f(\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})) < f(\mathbf{x}^{(0)}). \quad (3.56)$$

This means that the point $\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$ is an improvement over the point $\mathbf{x}^{(0)}$ if we are searching for a minimizer. To formulate an algorithm that implements the above idea, suppose that we are given a point $\mathbf{x}^{(k)}$. To find the next point $\mathbf{x}^{(k+1)}$, we start at $\mathbf{x}^{(k)}$ and move by an amount $-\alpha_k \nabla f(\mathbf{x}^{(k)})$, where α_k is a positive scalar called the step size. The above procedure leads to the following iterative algorithm:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}). \quad (3.57)$$

We refer to the above as a gradient descent algorithm (or simply a gradient algorithm). The gradient varies as the search proceeds, tending to zero as we approach the minimizer. We have the option of either taking very small steps and re-evaluating the gradient at every step, or we can take large steps each time. The first approach results in a laborious method of reaching the minimizer, whereas the second approach may result in a more zigzag path to the minimizer. The advantage of the second approach is a possibly fewer number of the gradient evaluations. Among many different methods that use the above philosophy the most popular is the method of steepest descent, which we discuss next.

The Method of Steepest Descent

The method of steepest descent is a gradient algorithm where the step size α_k is chosen to achieve the maximum amount of decrease of the objective function at each individual step. Specifically, α_k is chosen to minimize $\phi_k(\alpha) \triangleq f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$. In other words,

$$\alpha_k = \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)})). \quad (3.58)$$

To summarize, the steepest descent algorithm proceeds as follows: at each step, starting from the point $\mathbf{x}^{(k)}$, we conduct a line search in the direction $-\nabla f(\mathbf{x}^{(k)})$ until a minimizer, $\mathbf{x}^{(k+1)}$, is found. A typical sequence resulting from the method of steepest descent is depicted in Figure 3.19. Observe that the method of steepest descent moves in orthogonal steps, as stated in the following proposition.

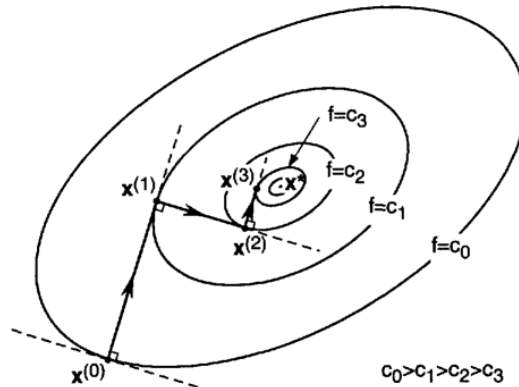


Figure 3.19. Typical sequence resulting from the method of steepest descent.

Theorem 3.10: If $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ is a steepest descent sequence for a given function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, then for each k the vector $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ is orthogonal to the vector $\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)}$.

Proof:

From the iterative formula of the method of steepest descent it follows that

$$\langle \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)} \rangle = \alpha_k \alpha_{k+1} \langle \nabla f(\mathbf{x}^{(k)}), \nabla f(\mathbf{x}^{(k+1)}) \rangle.$$

To complete the proof it is enough to show that

$$\langle \nabla f(\mathbf{x}^{(k)}), \nabla f(\mathbf{x}^{(k+1)}) \rangle = 0.$$

To this end, observe that α_k is a nonnegative scalar that minimizes $\phi_k(\alpha) \triangleq f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$. Hence,

$$\begin{aligned} 0 &= \phi'_k(\alpha_k) \\ &= \frac{d\phi_k}{d\alpha}(\alpha_k) \\ &= \langle \nabla f(\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})), -\nabla f(\mathbf{x}^{(k)}) \rangle \\ &= -\langle \nabla f(\mathbf{x}^{(k+1)}), \nabla f(\mathbf{x}^{(k)}) \rangle \end{aligned}$$

and the proof is completed. ■

The above proposition implies that $\nabla f(\mathbf{x}^{(k)})$ is parallel to the tangent plane to the level set $\{f(\mathbf{x}) = f(\mathbf{x}^{(k+1)})\}$ at $\mathbf{x}^{(k+1)}$. Note that as each new point is generated by the steepest descent algorithm, the corresponding value of the function f decreases in value, as stated below.

Theorem 3.11: If $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ is a steepest descent sequence for $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and if $\nabla f(\mathbf{x}^{(k)}) \neq 0$, then

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}). \quad (3.59)$$

Proof:

First recall that

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}),$$

where $\alpha_k \geq 0$ is the minimizer of

$$\phi_k(\alpha) = f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$$

over all $\alpha \geq 0$. Thus, for $\alpha \geq 0$, we have

$$\phi_k(\alpha_k) \leq \phi_k(\alpha).$$

By the chain rule,

$$\phi'_k(0) = \frac{d\phi_k}{d\alpha}(0) = -\langle \nabla f(\mathbf{x}^{(k)} - 0 \times \nabla f(\mathbf{x}^{(k)})), \nabla f(\mathbf{x}^{(k)}) \rangle = -\|\nabla f(\mathbf{x}^{(k)})\|^2 < 0$$

because $\nabla f(\mathbf{x}^{(k)}) \neq 0$ by assumption. Thus, $\phi'_k(0) < 0$ and this implies that there is an $\bar{\alpha} > 0$ such that $\phi_k(0) > \phi_k(\alpha)$ for all $\alpha \in (0, \bar{\alpha}]$. Hence,

$$f(\mathbf{x}^{(k+1)}) = \phi_k(\alpha_k) \leq \phi_k(\bar{\alpha}) < \phi_k(0) = f(\mathbf{x}^{(k)})$$

and the proof of the statement is completed. ■

In the above, we proved that the algorithm possesses the descent property: $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ if $\nabla f(\mathbf{x}^{(k)}) \neq 0$. In this case, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$. We can use the above as the basis for a stopping (termination) criterion for the algorithm.

The condition $\nabla f(\mathbf{x}^{(k+1)}) = 0$, however, is not directly suitable as a practical stopping criterion, because the numerical computation of the gradient will rarely be identically equal to zero. A practical stopping criterion is to check

if the norm $\|\nabla f(\mathbf{x}^{(k)})\|$ of the gradient is less than a prespecified threshold, in which case we stop. Alternatively, we may compute the absolute difference $|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})|$ between objective function values for every two successive iterations, and if the difference is less than some prespecified threshold, then we stop; that is, we stop when

$$|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})| < \varepsilon, \quad (3.60)$$

where $\varepsilon > 0$ is a prespecified threshold. Yet another alternative is to compute the norm $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$ of the difference between two successive iterates, and we stop if the norm is less than a prespecified threshold:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon. \quad (3.61)$$

Alternatively, we may check "relative" values of the above quantities; for example,

$$\frac{|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})|}{|f(\mathbf{x}^{(k)})|} < \varepsilon, \quad (3.62)$$

or

$$\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k)}\|} < \varepsilon. \quad (3.63)$$

The above two (relative) stopping criteria are preferable to the previous (absolute) criteria because the relative criteria are "scale-independent." For example, scaling the objective function does not change the satisfaction of the criterion $|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})|/|f(\mathbf{x}^{(k)})| < \varepsilon$. Similarly, scaling the decision variable does not change the satisfaction of the criterion $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|/\|\mathbf{x}^{(k)}\| < \varepsilon$. To avoid dividing by very small numbers, we can modify these stopping criteria as follows:

$$\frac{|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})|}{\max(1, |f(\mathbf{x}^{(k)})|)} < \varepsilon, \quad (3.64)$$

or

$$\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\max(1, \|\mathbf{x}^{(k)}\|)} < \varepsilon. \quad (3.65)$$

Note that the above stopping criteria are relevant to all the iterative algorithms we discuss in this part.

Example 3.6

We use the method of steepest descent to find the minimizer of

$$f(x_1, x_2, x_3) = (x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4.$$

The initial point is $\mathbf{x}^{(0)} = (4, 2, -1)^T$.

Solution

We perform three iterations. We find

$$\nabla f(\mathbf{x}) = (4(x_1 - 4)^3, 2(x_2 - 3), 14(x_3 + 5)^3)^T.$$

Hence,

$$\nabla f(\mathbf{x}^{(0)}) = (0, -2, 1024)^T.$$

and

$$\begin{aligned} \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)}) &= (4, 2, -1)^T - \alpha(0, -2, 1024)^T \\ &= (4, 2, -1)^T - (0, -2\alpha, 1024\alpha)^T \\ &= (4, 2 + 2\alpha, -1 - 1024\alpha)^T \end{aligned}$$

So that

$$\begin{aligned} f(\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})) &= f(4, 2 + 2\alpha, -1 - 1024\alpha) \\ &= (4 - 4)^4 + (2 + 2\alpha - 3)^2 + 4(-1 - 1024\alpha + 5)^4 \\ &= (0)^4 + (2\alpha - 1)^2 + 4(4 - 1024\alpha)^4 \end{aligned}$$

To compute $\mathbf{x}^{(1)}$, we need

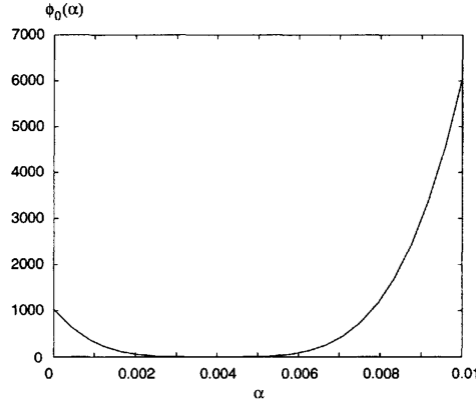


Figure 3.20. Plot of $\phi_0(\alpha)$ versus α

$$\begin{aligned}
 \alpha_0 &= \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})) \\
 &= \arg \min_{\alpha \geq 0} ((2\alpha - 1)^2 + 4(4 - 1024\alpha)^4) \\
 &= \arg \min_{\alpha \geq 0} \phi_0(\alpha).
 \end{aligned}$$

Using the secant method from the previous chapter, we obtain

$$\alpha_0 = 3.967 \times 10^{-3}.$$

For illustrative purpose, we show a plot of $\phi_0(\alpha)$ versus α in [figure 3.20](#).

Thus,

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha_0 \nabla f(\mathbf{x}^{(0)}) = (4.000, 2.008, -5.062)^T.$$

To find $\mathbf{x}^{(2)}$, we first determine

$$\nabla f(\mathbf{x}^{(1)}) = (0.000, -1.984, -0.003875)^T.$$

Next, we find α_1 , where

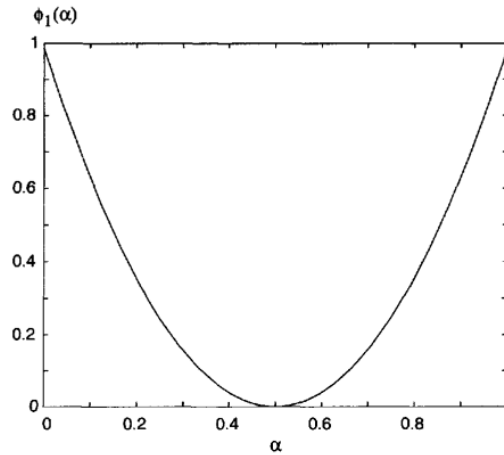


Figure 3.21. Plot of $\phi_1(\alpha)$ versus α

$$\begin{aligned}
 \alpha_1 &= \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(1)} - \alpha \nabla f(\mathbf{x}^{(1)})) \\
 &= \arg \min_{\alpha \geq 0} (0 + (2.008 + 1.984\alpha - 3)^2 + 4(-5.062 + 0.003875\alpha + 5)^4) \\
 &= \arg \min_{\alpha \geq 0} \phi_1(\alpha).
 \end{aligned}$$

Using the secant method again, we obtain $\alpha_1 = 0.5000$. [Figure 3.21](#) depicts a plot of $\phi_1(\alpha)$ versus α .

Thus,

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha_1 \nabla f(\mathbf{x}^{(1)}) = (4.000, 3.000, -5.060)^T.$$

To find $\mathbf{x}^{(3)}$ we first determine

$$\nabla f(\mathbf{x}^{(2)}) = (0.000, 0.000, -0.003525)^T.$$

and

$$\begin{aligned} \alpha_2 &= \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(2)} - \alpha \nabla f(\mathbf{x}^{(2)})) \\ &= \arg \min_{\alpha \geq 0} (0.000 + 0.000 + 4(-5.060 + 0.003525\alpha + 5)^4) \\ &= \arg \min_{\alpha \geq 0} \phi_2(\alpha) \end{aligned}$$

We proceed as in the previous iterations to obtain $\alpha_2 = 16.29$. A plot of $\phi_2(\alpha)$ versus α is shown in [figure 3.22](#). The value of $\mathbf{x}^{(3)}$ is

$$\mathbf{x}^{(3)} = (4.000, 3.000, -5.002)^T.$$

Note that the minimizer of f is $(4, 3, -5)^T$, and hence it appears that we have arrived at the minimizer in only three iterations.

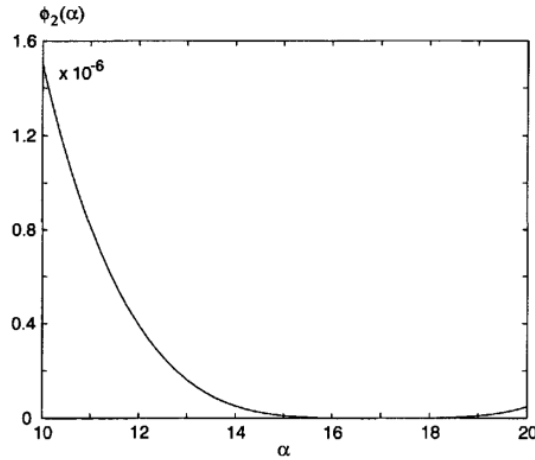


Figure 3.22. Plot of $\phi_2(\alpha)$ versus α .

Newton's Method

Recall that the method of steepest descent uses only first derivatives (gradients) in selecting a suitable search direction. This strategy is not always the most effective. If higher derivatives are used, the resulting iterative algorithm may perform better than the steepest descent method. Newton's method (sometimes called the Newton-Raphson method) uses first and second derivatives and indeed does perform better than the steepest descent method if the initial point is close to the minimizer. The idea behind this method is as follows. Given a starting point, we construct a quadratic approximation to the objective function that matches the first and second derivative values at that point. We then minimize the approximate (quadratic) function instead of the original objective function. We use the minimizer of the approximate function as the starting point in the next step and repeat the procedure iteratively. If the objective function is quadratic, then the approximation is exact, and the method yields the true minimizer in one step. If, on the other hand, the objective function is not quadratic, then the approximation will provide only an estimate of the position of the true minimizer. [Figure 3.23](#) illustrates the above idea.

We can obtain a quadratic approximation to the given twice continuously differentiable objection function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ using the Taylor series expansion of f about the current point $\mathbf{x}^{(k)}$, neglecting terms of order three and higher. We obtain

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &\approx f(\mathbf{x}^{(k)}) + \langle \mathbf{G}^{(k)}, (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) \rangle + \frac{1}{2} \langle (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) | \mathbf{H}_f(\mathbf{x}^{(k)}) | (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) \rangle \\ &\triangleq q(\mathbf{x}^{(k+1)}). \end{aligned} \tag{3.66}$$

where $\mathbf{G}^{(k)} = \nabla f(\mathbf{x}^{(k)})$. Hence, we have

$$0 = \nabla q(\mathbf{x}^{(k+1)}) = \mathbf{G}^{(k)} + \mathbf{H}_f(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}). \quad (3.67)$$

If $\mathbf{H}_f(\mathbf{x}^{(k)}) > 0$, then q achieves a minimum at

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{H}_f(\mathbf{x}^{(k)})^{-1} \mathbf{G}^{(k)}. \quad (3.68)$$

This recursive formula represents Newton's method.

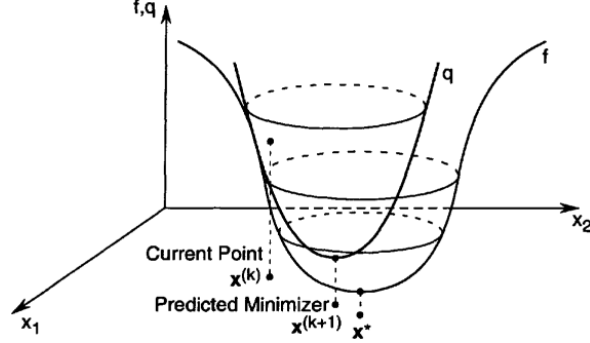


Figure 3.23. Quadratic approximation to the objective function using first and second derivatives

Example 3.7

Use Newton's method to minimize the Powell function:

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

Solution

Use as the starting point $\mathbf{x}^{(0)} = (3, -1, 0, 1)^T$. Perform three iterations.

Note that $f(\mathbf{x}^{(k)}) = 215$. We have

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 2(x_1 + 10x_2) + 40(x_1 - x_4)^3 \\ 20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3 \\ 10(x_3 - x_4) - 8(x_2 - 2x_3)^3 \\ -10(x_3 - x_4) - 40(x_1 - x_4)^3 \end{pmatrix},$$

and $\mathbf{H}_f(\mathbf{x})$ is given by

$$\begin{pmatrix} 2 + 120(x_1 - x_4)^2 & 20 & 0 & -120(x_1 - x_4)^2 \\ 20 & 200 + 12(x_2 - 2x_3)^2 & -24(x_2 - 2x_3)^2 & 0 \\ 0 & -24(x_2 - 2x_3)^2 & 10 + 48(x_2 - 2x_3)^2 & -10 \\ -120(x_1 - x_4)^2 & 0 & -10 & 10 + 120(x_1 - x_4)^2 \end{pmatrix}$$

Iteration 1.

$$\begin{aligned} \mathbf{g}^{(0)} &= [306, -144, -2, -310]^T, \\ \mathbf{F}(\mathbf{x}^{(0)}) &= \begin{bmatrix} 482 & 20 & 0 & -480 \\ 20 & 212 & -24 & 0 \\ 0 & -24 & 58 & -10 \\ -480 & 0 & -10 & 490 \end{bmatrix}, \\ \mathbf{H}_f(\mathbf{x}^{(0)})^{-1} &= \begin{pmatrix} .1126 & -.0089 & .0154 & .1106 \\ -.0089 & .0057 & .0008 & -.0087 \\ .0154 & .0008 & .0203 & .0155 \\ .1106 & -.0087 & .0155 & .1107 \end{pmatrix}, \\ \mathbf{H}_f(\mathbf{x}^{(0)})^{-1} \mathbf{G}^{(0)} &= (1.4127, -0.8413, -0.2540, 0.7460)^T. \end{aligned}$$

Hence,

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} - \mathbf{H}_f(\mathbf{x}^{(0)})^{-1} \mathbf{G}^{(0)} = (1.5873, -0.1587, 0.2540, 0.2540)^T, \\ f(\mathbf{x}^{(1)}) &= 31.8. \end{aligned}$$

Iteration 2.

$$\mathbf{H}_f(\mathbf{x}^{(1)}) = \begin{pmatrix} 215.3 & 20 & 0 & -213.3 \\ 20 & 205.3 & -10.67 & 0 \\ 0 & -10.67 & 31.34 & -10 \\ -213.3 & 0 & -10 & 223.3 \end{pmatrix},$$

$$\mathbf{H}_f(\mathbf{x}^{(1)})^{-1} \mathbf{G}^{(1)} = (0.5291, -0.0529, -0.0846, 0.0846)^T$$

Hence,

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \mathbf{H}_f(\mathbf{x}^{(1)})^{-1} \mathbf{G}^{(1)} = (1.0582, -0.1058, 0.1694, 0.1694)^T,$$

$$f(\mathbf{x}^{(2)}) = 6.28.$$

Iteration 3.

$$\mathbf{G}^{(2)} = (28.09, -0.3475, 0.7031, -28.08)^T,$$

$$\mathbf{H}_f(\mathbf{x}^{(2)}) = \begin{pmatrix} 96.80 & 20 & 0 & -94.80 \\ 20 & 202.4 & -4.744 & 0 \\ 0 & -4.744 & 19.49 & -10 \\ -94.80 & 0 & -10 & 104.80 \end{pmatrix},$$

$$\mathbf{x}^{(3)} = (0.7037, -0.0704, 0.1121, 0.1111)^T,$$

$$f(\mathbf{x}^{(3)}) = 1.24.$$

3.6 Mathematica Built in Functions

<code>D[f, {x₁, x₂, ...}]</code>	the gradient of a scalar function
<code>D[f, {x₁, x₂, ...}, 2]</code>	the Hessian matrix for f
<code>D[f, {x₁, x₂, ...}, n]</code>	the n th-order Taylor series coefficient

<code>PositiveDefiniteMatrixQ[m]</code>	gives True if m is explicitly positive definite, and False otherwise.
<code>NegativeDefiniteMatrixQ[m]</code>	gives True if m is explicitly negative definite, and False otherwise.
<code>PositiveSemidefiniteMatrixQ[m]</code>	gives True if m is explicitly positive semidefinite, and False otherwise.
<code>NegativeSemidefiniteMatrixQ[m]</code>	gives True if m is explicitly negative semidefinite, and False otherwise.

<code>Eigenvalues[m]</code>	gives a list of the eigenvalues of the square matrix m.
<code>Eigenvalues[m, k]</code>	gives the first k eigenvalues of m.
<code>Eigenvectors[m]</code>	gives a list of the eigenvectors of the square matrix m.
<code>Eigenvectors[m, k]</code>	gives the first k eigenvectors of m.
<code>Det[m]</code>	gives the determinant of the square matrix m.
<code>Inverse[m]</code>	gives the inverse of a square matrix m.

`ContourPlot[f, {x, xmin, xmax}, {y, ymin, ymax}]` generates a contour plot of f as a function of x and y .

`ContourPlot3D[f, {x, xmin, xmax}, {y, ymin, ymax}, {z, zmin, zmax}]` produces a three-dimensional contour plot of f as a function of x , y , and z .

Example 3.8

```

Input      : Grad[f[x, y, z], {x, y, z}]
Output     : {f(1,0,0)[x,y,z], f(0,1,0)[x,y,z], f(0,0,1)[x,y,z]}

Input      :
            (* The gradient in two dimensions: *)
            Grad[Sin[x^2 + y^2], {x, y}]
Output     : {2 x Cos[x^2 + y^2], 2 y Cos[x^2 + y^2]}

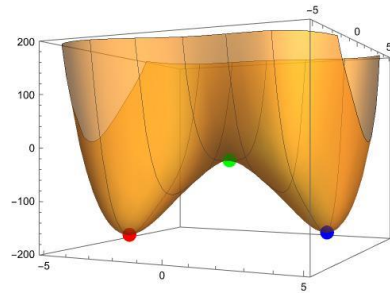
Input      : (* Compute the Hessian of a scalar function: *)
            grad = Grad[x*y*z, {x, y, z}]
            Grad[grad, {x, y, z}]
            MatrixForm [Grad[grad, {x, y, z}]]
Output     : {y z, x z, x y}
Output     : {{0, z, y}, {z, 0, x}, {y, x, 0}}
Output     :  $\begin{pmatrix} 0 & z & y \\ z & 0 & x \\ y & x & 0 \end{pmatrix}$ 

Input      : (* Find the critical points of a function of two variables: *)
            f[x_, y_] := x^4 + y^4 - 20 x^2 - 10 x y - 25
            grad = Grad[f[x, y], {x, y}]
            sol = NSolve[grad == {0, 0}, {x, y}, Reals]
Output     : {-40 x + 4 x^3 - 10 y, -10 x + 4 y^3}
Output     : {{x -> -3.39162, y -> -2.03915}, {x -> 3.39162, y -> 2.03915}, {x -> 0., y -> 0.}}

Input      : (* Compute the signs of  $\frac{\partial^2 f(x,y)}{\partial x^2}$  and the Hessian determinant: *)
            hessian = Grad[grad, {x, y}]
            Sign[hessian[[1, 1]] /. sol]
            Sign[Det[hessian] /. sol]
Output     : {{-40 + 12 x^2, -10}, {-10, 12 y^2}}
Output     : {1, 1, -1}
Output     : {1, 1, -1}

Input      : (* By the second derivative test, the first two points—red and blue in
            the plot—are minima, and the third—green in the plot—is a saddle point: *)
            Show[
            Plot3D[
                f[x, y], {x, -5, 5}, {y, -5, 5},
                PlotRange -> 200,
                ClippingStyle -> None,
                Mesh -> {5, 0},
                PlotStyle -> Opacity[.65],
                BoxRatios -> {1, 1, 3/4},
                ViewPoint -> {1.2, -2.5, 0}
            ],
            Graphics3D[
                {PointSize[.04],
                 Riffle[{Red, Blue, Green},
                     Point[{x, y, f[x, y]}] /. sol}}]
            ]
Output     :

```



```

Input      :
            (*Test if a 2x2 real matrix is explicitly positive definite:*)
            m =  $\begin{pmatrix} 5 & -1 \\ -1 & 4 \end{pmatrix}$ 
Input      : PositiveDefiniteMatrixQ[m]
Output     : True

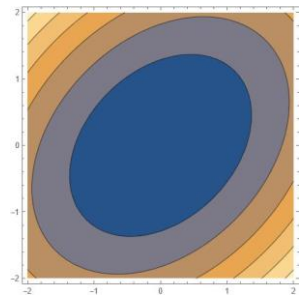
Input      :
            (*Consider a real, positive definite 2x2 matrix and its associated real
            quadratic q= xT.m.x:*)
            m =  $\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$ 
            PositiveDefiniteMatrixQ[m]
Output     : True

Input      : Eigenvalues[{{1.1, 2.2, 3.25}, {0.76, 4.6, 5}, {0.1, 0.1, 6.1}}]
Output     : {6.60674, 4.52536, 0.667901}

Input      : Eigenvectors[{{1.1, 2.2, 3.25}, {0.76, 4.6, 5}, {0.1, 0.1, 6.1}}]
Output     : {{0.48687, 0.833694, 0.260598}, {-0.479424, -0.873368,
            0.085911}, {0.985096, -0.171352, -0.0149803}}

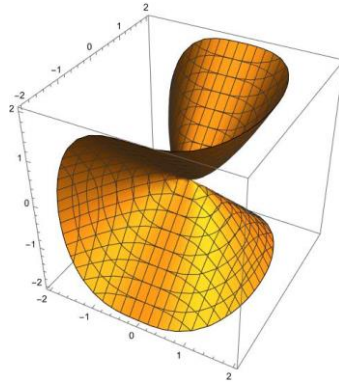
Input      : Det[{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}]
Output     : 0

Input      : Inverse[{{1.4, 2}, {3, -6.7}}]
Output     : {{0.435631, 0.130039}, {0.195059, -0.0910273}}

Input      : q[x_, y_] := {x, y} . m . {x, y}
            (*Because m is positive definite, the level sets are ellipses:*)
            ContourPlot[q[x, y], {x, -2, 2}, {y, -2, 2}]
Output     :
            

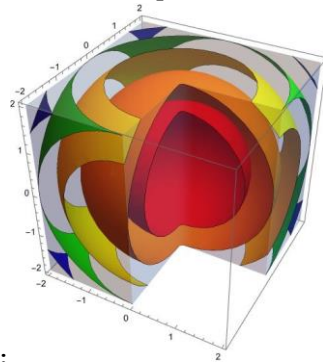
Input      : ContourPlot3D[x^3 + y^2 - z^2 == 0, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]
Output     :

```

Input : `ContourPlot3D[x^2 + y^2 + z^2, {x, -2, 2}, {y, -2, 2}, {z, -2, 2},
Contours -> 5, RegionFunction -> Function[{x, y, z}, x < 0 || y > 0],
ContourStyle -> {Red, Orange, Yellow, Green, Blue}, Mesh -> None]`

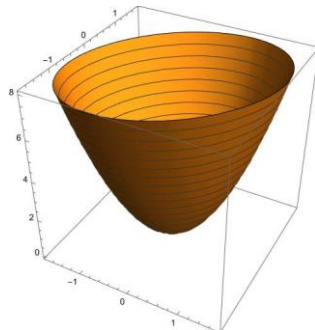
Output



:

Input : (*The plot of q will be an upward-facing elliptic paraboloid:*)
`Plot3D[q[x, y], {x, y} \[Element]
RotationTransform[\[Pi]/4][Disk[{0, 0}, {2, Sqrt[2]}]]],
MeshFunctions -> {#3 &}, BoxRatios -> 1]`

Output



<code>FindMinimum[f, {{x, x0}, {y, y0}, ...}]</code>	searches for a local minimum in a function of several variables.
<code>FindMinimumPlot[f, {{x, xst}, {y, yst}}]</code>	plots the steps and the points at which the bivariate function f and any of its derivatives are evaluated, superimposed on a contour plot of f as a function of x and y .
<code>Minimize[f, {x, y, ...}]</code>	minimizes f exactly with respect to x, y, \dots .
<code>NMinimize[f, {x, y, ...}]</code>	minimizes f numerically with respect to x, y, \dots .

Example 3.10

```

Input      : << Optimization`UnconstrainedProblems`
Input      :
Plot3D[Sin[x] Sin[2 y], {x, -3, 3}, {y, -3, 3},
  ColorFunction -> "Rainbow", PlotLegends -> BarLegend[Automatic]]

ContourPlot[Sin[x] Sin[2 y], {x, -3, 3}, {y, -3, 3},
  PlotLegends -> Automatic, Contours -> 10, ColorFunction -> "Rainbow"]

FindMinimum[Sin[x] Sin[2 y], {{x, 2}, {y, 2}}]

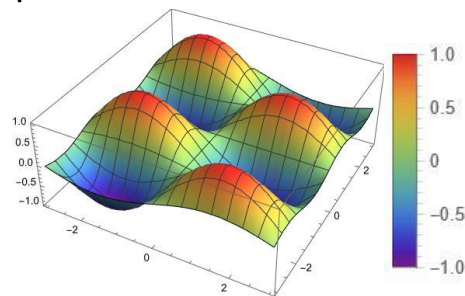
FindMinimumPlot[Sin[x] Sin[2 y], {{x, 2}, {y, 2}}, Method -> "Newton"]

FindMinimumPlot[Sin[x] Sin[2 y], {{x, 2}, {y, 2}},
  Method -> "QuasiNewton"]

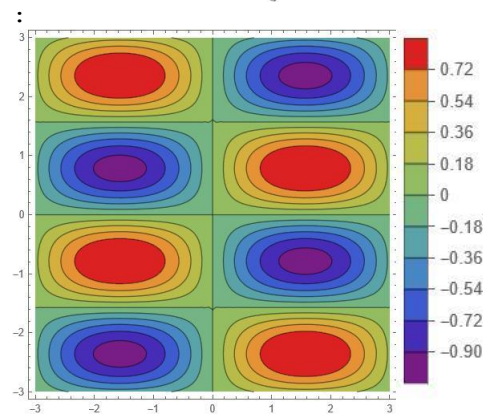
FindMinimumPlot[Sin[x] Sin[2 y], {{x, 2}, {y, 2}},
  Method -> "PrincipalAxis"]

```

Output :

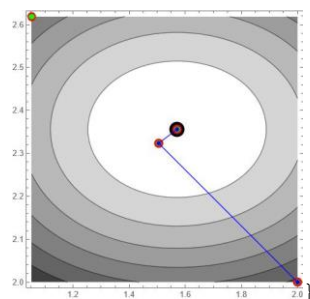


Output :

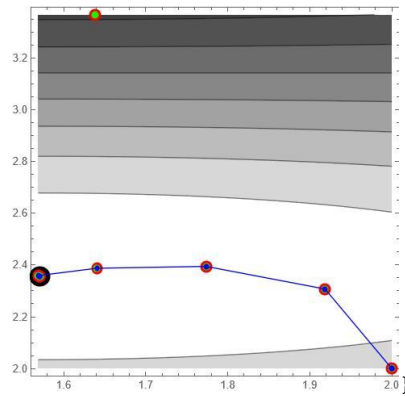


Output : {-1., {x -> 1.5708, y -> 2.35619}}

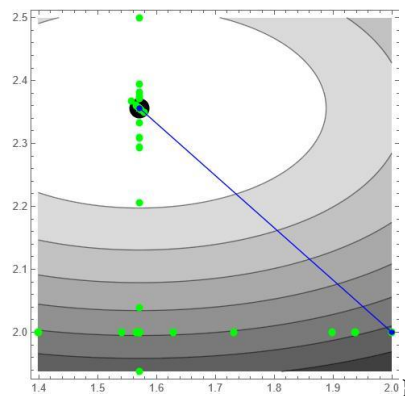
Output : {{-1., {x -> 1.5708, y -> 2.35619}}, {"Steps" -> 3, "Function" -> 5, "Gradient" -> 5},



Output : {{-1., {x -> 1.5708, y -> 2.35619}}, {"Steps" -> 7, "Function" -> 9, "Gradient" -> 9},



Output : {{-1., {x -> 1.5708, y -> 2.35619}}, {"Steps" -> 2, "Function" -> 72},



Example 3.11

Input :

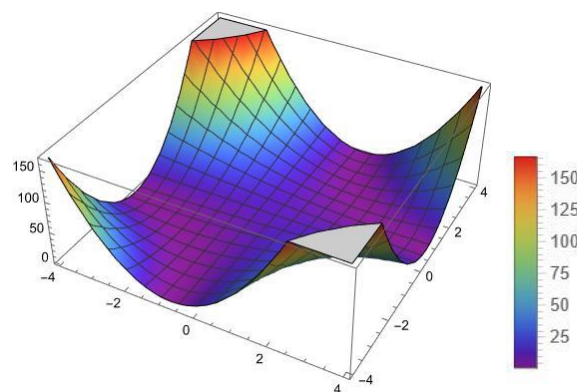
```
Plot3D[(x y - 3)^2 + 1, {x, -4, 4}, {y, -4, 4},
  ColorFunction -> "Rainbow", PlotLegends -> BarLegend[Automatic]]

ContourPlot[(x y - 3)^2 + 1, {x, -4, 4}, {y, -4, 4},
  PlotLegends -> Automatic, Contours -> 10, ColorFunction -> "Rainbow"]

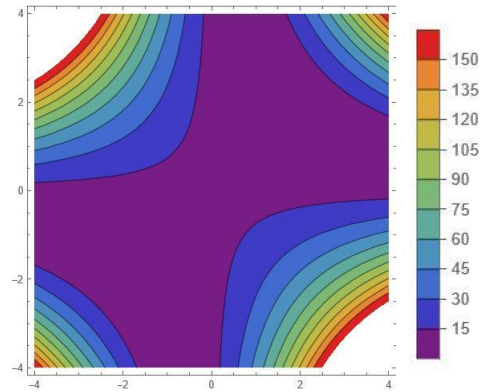
Minimize[(x y - 3)^2 + 1, {x, y}]

FindMinimumPlot[(x y - 3)^2 + 1, {{x, -1}, {y, -1}},
  Method -> "Newton"]
```

Output :

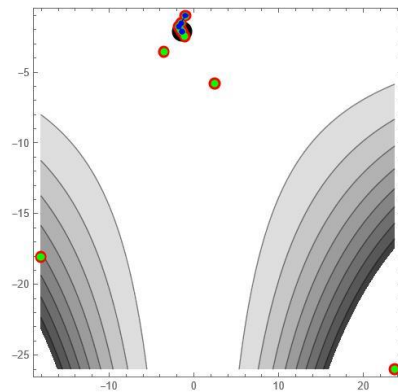


Output :



Output : {1, {x -> -1, y -> -3}}

Output : {"Steps" -> 5, "Function" -> 12, "Gradient" -> 12},



Example 3.12

```

Input
:
h = Cos[x^2 - 3 y] + Sin[x^2 + y^2]

Plot3D[h, {x, -3, 3}, {y, -3, 3}, ColorFunction -> "Rainbow",
  PlotLegends -> BarLegend[Automatic]]

ContourPlot[h, {x, -3, 3}, {y, -3, 3}, PlotLegends -> Automatic,
  Contours -> 10, ColorFunction -> "Rainbow"]

Minimize[h, {x, y}]

FindMinimum[h, {{x, 1}, {y, 1}}, Method -> "Newton"]

FindMinimumPlot[h, {{x, 1}, {y, 1}}, Method -> "Newton"]

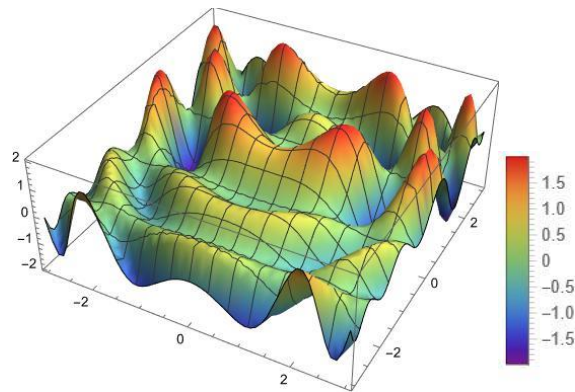
FindMinimumPlot[h, {{x, 1}, {y, 1}}, Method -> "QuasiNewton"]

FindMinimumPlot[h, {{x, 1}, {y, 1}}, Method -> "ConjugateGradient"]

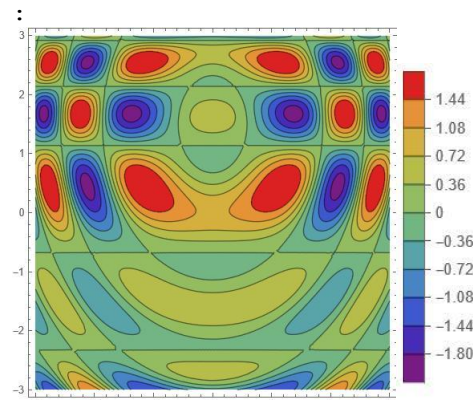
FindMinimumPlot[h, {{x, 1}, {y, 1}}, Method -> "PrincipalAxis"]

Output
: Cos[x^2 - 3 y] + Sin[x^2 + y^2]
Output
:

```



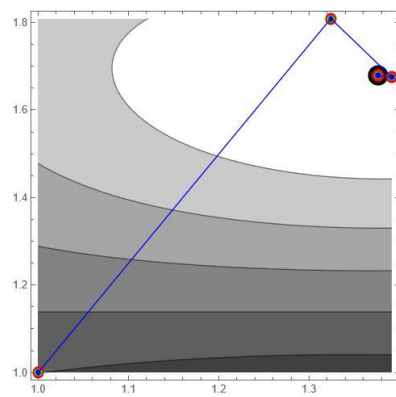
Output



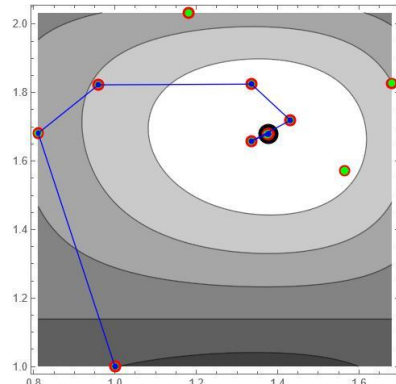
Output : $\{-2., \{x \rightarrow -\sqrt{1/2} (-9 - 2 \sqrt{\pi}) + 9 \sqrt{1 + 82 \sqrt{\pi}}\},$
 $y \rightarrow 3/2 (-1 + \sqrt{1 + 82 \sqrt{\pi}})\}$

Output : $\{-2., \{x \rightarrow 1.37638, y \rightarrow 1.67868\}\}$

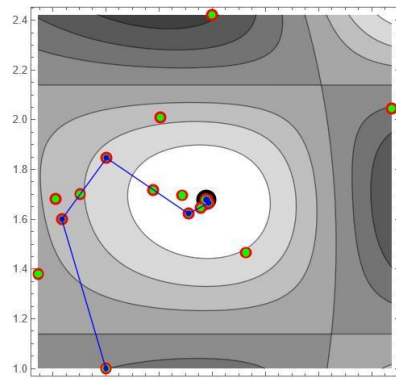
Output : $\{\{-2., \{x \rightarrow 1.37638, y \rightarrow 1.67868\}\}, \{"Steps" \rightarrow 5, "Function" \rightarrow 6,$
 $"Gradient" \rightarrow 6\},$



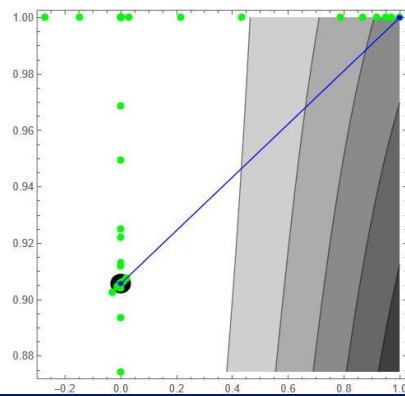
Output : $\{\{-2., \{x \rightarrow 1.37638, y \rightarrow 1.67868\}\}, \{"Steps" \rightarrow 9, "Function" \rightarrow 13,$
 $"Gradient" \rightarrow 13\},$



Output : `{{-2., {x -> 1.37638, y -> 1.67868}}, {"Steps" -> 10, "Function" -> 32, "Gradient" -> 32},`



Output : `{{-0.179902, {x -> 4.94463*10^-9, y -> 0.905726}}, {"Steps" -> 1, "Function" -> 56},`



Resource Functions

HessianMatrix: Compute the Hessian matrix of a function with respect to a list of variables

<code>ResourceFunction["HessianMatrix"]</code>	computes the Hessian matrix of the
<code>[expr, {var1, var2, ...}]</code>	expression <code>expr</code> with respect to the
	given variables.

HessianDeterminant: Compute the Hessian determinant of a function with respect to a list of variables

<code>ResourceFunction["HessianDeterminant"]</code>	computes the determinant of the
<code>[expr, {var1, var2, ...}]</code>	Hessian matrix of the expression
	<code>expr</code> with respect to the given
	variables.

Example 3.9

Input	: <code>ResourceFunction["HessianMatrix"][x Cos[y], {x, y}]</code>
Output	: <code>{{0, -Sin[y]}, {-Sin[y], -x Cos[y]}}</code>
Input	: <code>ResourceFunction["HessianDeterminant"][Sin[x] Cos[y], {x, y}]</code>
Output	: <code>1/2 (-Cos[2 x] + Cos[2 y])</code>