

# La Gradiente de una Función

Alumno: Ronald Wilder Incacutipa Muñuico

Docente: Ing. Torres Cruz Fred

Programación Numérica – FINESI  
Universidad Nacional del Altiplano

## Definición de la Gradiente

La **gradiente** de una función escalar  $f$  es un vector que contiene sus derivadas parciales. Si  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , entonces su gradiente se define como:

$$\nabla f(x_1, x_2, \dots, x_n) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$

Este vector apunta en la dirección de **mayor incremento** de la función. En métodos numéricos, se usa para el *descenso del gradiente*, donde los valores de las variables se actualizan de la siguiente forma:

$$x_{i+1} = x_i - \eta \nabla f(x_i)$$

donde  $\eta$  es la **tasa de aprendizaje** o tamaño de paso.

## Ejemplo 1: Función de una variable

Sea la función:

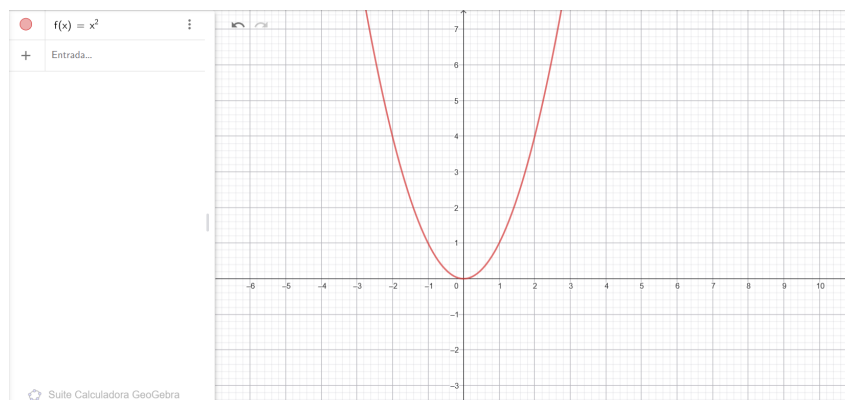
$$f(x) = x^2$$

Por tanto:

$$\nabla f(x_i) = x_{i-1} - \eta \frac{df(x_{i-1})}{dx}$$

Con  $\eta = 0.1$

Con gráfica:



Su derivada es:

$$f'(x) = 2x$$

A continuación, se presenta el desarrollo del método del gradiente con una tasa de aprendizaje  $\eta = 0,1$  y un punto inicial  $x_0 = 5$  durante 15 iteraciones.

**Código en R:**

```
1  # Definimos la función y su derivada
2  f <- function(x) { x^2 }          # f(x) = x^2
3  f_deriv <- function(x) { 2*x }    # f'(x) = 2x
4
5  # Parámetros
6  x0 <- 5                          # Valor inicial
7  eta <- 0.1                       # Tasa de aprendizaje
8  n_iter <- 15                     # Número de iteraciones
9
10 # Inicializamos vectores
11 x_vals <- numeric(n_iter)
12 f_vals <- numeric(n_iter)
13 deriv_vals <- numeric(n_iter)
14
15 # Asignamos el primer valor
16 x_vals[1] <- x0
17 f_vals[1] <- f(x0)
18 deriv_vals[1] <- f_deriv(x0)
19
20 # Calculamos iterativamente el gradiente
21 for (i in 2:n_iter) {
22   x_new <- x_vals[i-1] - eta * f_deriv(x_vals[i-1])
23   x_vals[i] <- x_new
24   f_vals[i] <- f(x_new)
25   deriv_vals[i] <- f_deriv(x_new)
26 }
27
28 # Graficamos: eje X son los valores reales de x
29 plot(x_vals, f_vals, type="l", col="blue", lwd=2,
30      xlab="Valores de x (actualizaciones del gradiente)",
31      ylab="Valor",
32      main="Función, Derivada y Gradiente Descendente (eje X = valores de x)")
33 lines(x_vals, deriv_vals, col="red", lwd=2)
34 lines(x_vals, x_vals, col="green", lwd=2)
35
36 legend("topright",
37       legend=c("f(x)", "f'(x)", "x_i (valores del gradiente)"),
38       col=c("blue", "red", "green"),
39       lwd=2)
```

**Resultados de las iteraciones:**

Iteración	$x_i$	$f(x_i)$	$\frac{df(x)}{dx} = 2x_i$	$x_{i+1}$
1	5.000	25.000	10.000	4.000
2	4.000	16.000	8.000	3.200
3	3.200	10.240	6.400	2.560
4	2.560	6.554	5.120	2.048
5	2.048	4.194	4.096	1.638
6	1.638	2.685	3.277	1.311
7	1.311	1.718	2.621	1.049
8	1.049	1.099	2.097	0.839
9	0.839	0.704	1.678	0.671
10	0.671	0.450	1.342	0.537
11	0.537	0.288	1.074	0.430
12	0.430	0.184	0.859	0.344
13	0.344	0.118	0.687	0.275
14	0.275	0.076	0.550	0.220
15	0.220	0.048	0.440	0.176

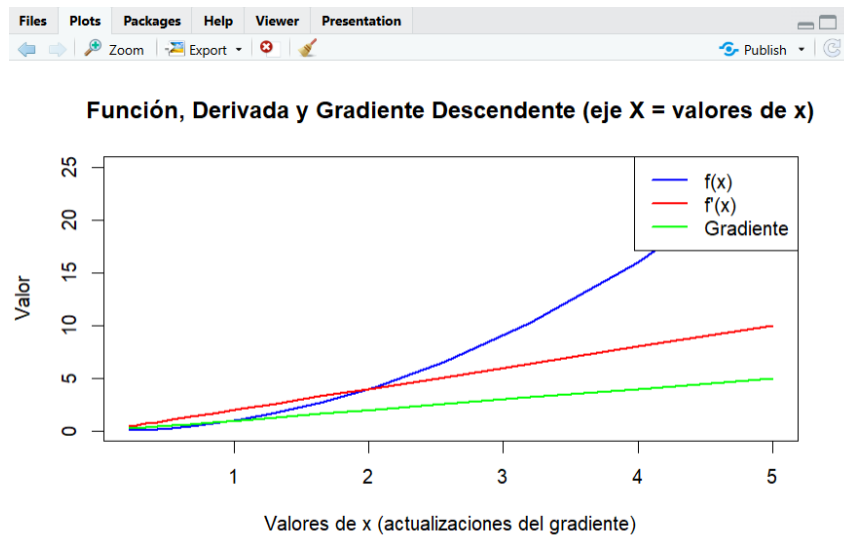


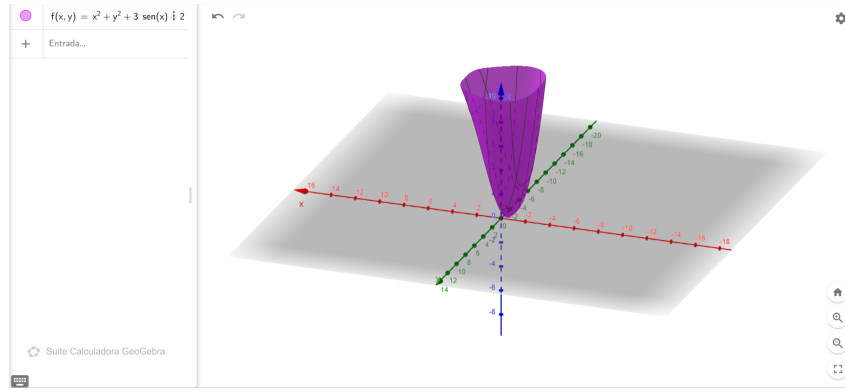
Figura 1. Descenso del gradiente para  $f(x) = x^2$

## Ejemplo 2: Función de dos variables

Sea la función:

$$f(x, y) = x^2 + y^2 + 3 \sin(x) + 2 \cos(y)$$

Con gráfica:



Las derivadas parciales son:

$$\frac{\partial f}{\partial x} = 2x + 3 \cos(x), \quad \frac{\partial f}{\partial y} = 2y - 2 \sin(y)$$

Por tanto:

$$\nabla f(x, y) = (2x + 3 \cos(x), 2y - 2 \sin(y))$$

Tambien:

$$x_{i+1} = x_{i-1} - \eta \frac{df(x_{i-1})}{dx} \quad y_{i+1} = y_{i-1} - \eta \frac{df(y_{i-1})}{dy}$$

Con  $\eta = 0.1$

**Resultados de las iteraciones:**

Iteración	$x_i$	$y_i$	$f(x_i, y_i)$	$\frac{\partial f}{\partial x}$	$\frac{\partial f}{\partial y}$	$x_{i+1}, y_{i+1}$
1	5.000	3.000	29.143	10.851	5.718	3.915, 2.428
2	3.915	2.428	17.615	5.683	3.548	3.347, 2.073
3	3.347	2.073	13.925	3.756	2.394	2.971, 1.834
4	2.971	1.834	12.179	2.986	1.737	2.672, 1.660
5	2.672	1.660	11.076	2.669	1.329	2.406, 1.527
6	2.406	1.527	10.221	2.588	1.057	2.147, 1.422
7	2.147	1.422	9.443	2.660	0.866	1.881, 1.335
8	1.881	1.335	8.644	2.846	0.726	1.596, 1.263
9	1.596	1.263	7.748	3.116	0.619	1.285, 1.201
10	1.285	1.201	6.693	3.416	0.537	0.943, 1.147
11	0.943	1.147	5.455	3.648	0.471	0.578, 1.100
12	0.578	1.100	4.091	3.669	0.417	0.211, 1.058
13	0.211	1.058	2.774	3.356	0.373	-0.124, 1.021
14	-0.124	1.021	1.731	2.728	0.337	-0.397, 0.987
15	-0.397	0.987	1.074	1.972	0.305	-0.594, 0.957

**Código en R:**

```

1 # =====
2 # DESCENSO DEL GRADIENTE EN 3D CON PLOT3D (CORREGIDO)
3 # f(x, y) = x^2 + y^2 + 3*sin(x) + 2*cos(y)
4 # =====
5
6 library(plot3D)

```

```

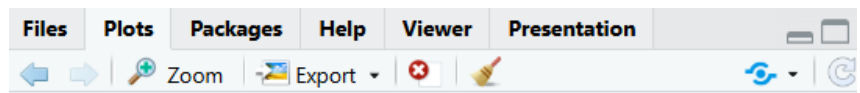
7
8 # Función y gradiente
9 f <- function(x, y) {
10   x^2 + y^2 + 3*sin(x) + 2*cos(y)
11 }
12
13 grad <- function(x, y) {
14   df_dx <- 2*x + 3*cos(x)
15   df_dy <- 2*y - 2*sin(y)
16   c(df_dx, df_dy)
17 }
18
19 # Parámetros
20 x0 <- 5
21 y0 <- 3
22 eta <- 0.1
23 n_iter <- 25
24
25 # Inicialización
26 x_vals <- numeric(n_iter)
27 y_vals <- numeric(n_iter)
28 f_vals <- numeric(n_iter)
29 dfx_vals <- numeric(n_iter)
30 dfy_vals <- numeric(n_iter)
31
32 x_vals[1] <- x0
33 y_vals[1] <- y0
34 f_vals[1] <- f(x0, y0)
35 g <- grad(x0, y0)
36 dfx_vals[1] <- g[1]
37 dfy_vals[1] <- g[2]
38
39 # Descenso del gradiente
40 for (i in 2:n_iter) {
41   g <- grad(x_vals[i-1], y_vals[i-1])
42   x_vals[i] <- x_vals[i-1] - eta * g[1]
43   y_vals[i] <- y_vals[i-1] - eta * g[2]
44   f_vals[i] <- f(x_vals[i], y_vals[i])
45   dfx_vals[i] <- g[1]
46   dfy_vals[i] <- g[2]
47 }
48
49 # =====
50 # Tabla de iteraciones
51 # =====
52 tabla_iteraciones <- data.frame(
53   Iteracion = 1:n_iter,
54   x_i = round(x_vals, 6),
55   y_i = round(y_vals, 6),
56   `f(x_i, y_i)` = round(f_vals, 6),
57   `f/x` = round(dfx_vals, 6),
58   `f/y` = round(dfy_vals, 6),

```

```

59   x_i1 = c(round(x_vals[-1], 6), NA),
60   y_i1 = c(round(y_vals[-1], 6), NA)
61 )
62
63 print(tabla_iteraciones)
64
65 # =====
66 # Superficie 3D usando plot3D::surf3D
67 # =====
68
69 # Crear la malla correctamente (X, Y matrices)
70 x_seq <- seq(-6, 6, length.out = 100)
71 y_seq <- seq(-6, 6, length.out = 100)
72 malla <- mesh(x_seq, y_seq)
73 X <- malla$x
74 Y <- malla$y
75 Z <- f(X, Y)
76
77 # Gráfica 3D de la superficie
78 surf3D(
79   x = X, y = Y, z = Z,
80   colkey = FALSE,
81   theta = 45, phi = 25, expand = 0.6,
82   col = jet.col(100),
83   border = "gray40",
84   main = "Descenso del Gradiente - f(x, y)",
85   xlab = "x", ylab = "y", zlab = "f(x, y)"
86 )
87
88 # Trayectoria del descenso
89 lines3D(
90   x_vals, y_vals, f_vals,
91   col = "blue", lwd = 2, add = TRUE
92 )
93
94 # Puntos del recorrido
95 points3D(
96   x_vals, y_vals, f_vals,
97   col = "red", pch = 19, cex = 1.2, add = TRUE
98 )
99
100 # Punto inicial y final
101 points3D(x_vals[1], y_vals[1], f_vals[1], col = "orange", pch = 17, cex = 2, add = TRUE)
102 points3D(tail(x_vals, 1), tail(y_vals, 1), tail(f_vals, 1), col = "darkgreen", pch = 8, cex = 2, add = TRUE)
103
104 # Leyenda
105 legend("topright",
106       legend = c("Trayectoria", "Puntos", "Inicio", "Final"),
107       col = c("blue", "red", "orange", "darkgreen"),
108       pch = c(NA, 19, 17, 8),
109       lwd = 2, bg = "white")
110

```



## Descenso del Gradiente - $f(x, y)$

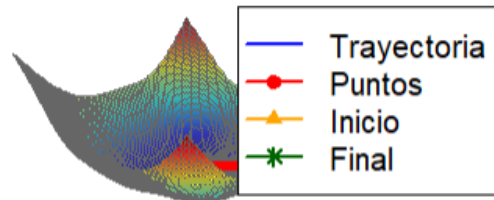


Figura 2. Descenso del gradiente para  $f(x, y) = x^2 + y^2 + 3 \sin(x) + 2 \cos(y)$